

Parametrische Modellordnungsreduktion für hierarchische selbstoptimierende Systeme

zur Erlangung des akademischen Grades eines
DOKTORS DER INGENIEURWISSENSCHAFTEN (Dr.-Ing.)
der Fakultät Maschinenbau
der Universität Paderborn

genehmigte
DISSERTATION

von
Dipl.-Math. Martin Krüger
aus Bad Oeynhausen

Tag des Kolloquiums:	8. November 2013
Referent:	Prof. Dr.-Ing. habil. Ansgar Trächtler
Korreferent:	Prof. Dr. Michael Dellnitz

Zusammenfassung

In dieser Arbeit wird eine neuartige Methode zur parametrischen Modellordnungsreduktion paretooptimaler Systeme vorgestellt. Mit dieser Methode können die komplexen Modelle, die im Rahmen selbstoptimierender Systeme auftreten, gezielt vereinfacht werden. Das herausragende Merkmal der entwickelten Methodik besteht in der engen Verzahnung der Verfahren der parametrischen Modellordnungsreduktion mit der hierarchischen Optimierung auf der einen Seite und dem Konzept der hierarchischen Strukturierung und Modellierung mechatronischer Systeme auf der anderen Seite.

Es werden zwei Varianten der parametrischen Modellordnungsreduktion betrachtet. Die manuelle rationale Interpolation und die Matrix Interpolation in Kombination mit der \mathcal{H}_2 -optimalen tangentialen Interpolation. Beide profitieren erheblich von der neu entstandenen Methode zur Interpolation paretooptimaler Systeme. Mit Hilfe geeigneter Parametrierungen und einer automatisiert durchführbaren Diskretisierung der Paretomenge werden die optimalen Systemkonfigurationen zusammen mit dem zugehörigen System zu einer Einheit gekapselt. Zudem verringert sich die Komplexität der parametrischen Reduktion, da die Anzahl beizubehaltender Parameter nur von der Anzahl der Zielfunktionen abhängt.

Am Beispiel des Feder-Neige-Prüfstands, einem Prüfstand für die aktive Federung des Schienenverkehrssystems RailCab, kann die hohe Approximationsgüte der Reduktion nachgewiesen werden.

Abstract

Parametric model-order reduction for the reduction of Pareto optimal systems is presented within this thesis. The developed method can be used to simplify complex models which describe the dynamical behavior of self-optimizing systems. The close interrelation of parametric model-order reduction with both hierarchical optimization as well as the structuring concept and hierarchical modeling of mechatronic systems is an outstanding feature of the proposed method.

Two types of parametric model-order reduction are considered. A particular Arnoldi algorithm for manual rational interpolation has been implemented on the one hand. On the other hand, a combination of matrix interpolation and \mathcal{H}_2 -optimal tangential interpolation has been investigated. Both types considerably benefit from a method for interpolation of Pareto optimal systems which has been developed within this thesis. Optimal system configurations and corresponding dynamical systems are encapsulated by means of suitable parameterizations and automatic discretization of the Pareto set. Additionally, the complexity of the reduction problem itself is reduced as the number of parameters only depends on the number of objective functions.

A test rig of an active suspension system representing the active suspension of the RailCab serves as an application example. The reduced models computed by means of the proposed method possess a very small reduction error.

Danksagung

Die hier vorliegende Arbeit ist im Rahmen meiner Tätigkeit als wissenschaftlicher Mitarbeiter der Fachgruppe Regelungstechnik und Mechatronik des Heinz Nixdorf Instituts im Sonderforschungsbereich 614 – Selbstoptimierende Systeme des Maschinenbaus entstanden. In dieser Zeit wurde ich in verschiedenster Weise unterstützt, was mich sehr dankbar macht.

Ich danke Herrn Prof. Dr.-Ing. Ansgar Trächtler für das mir entgegen gebrachte Vertrauen und die Ermöglichung dieser Arbeit. Er ließ mir große Freiheiten in der wissenschaftlichen Ausarbeitung der Thematik, stand aber gleichzeitig stets als Ansprechpartner zur Verfügung und gab immer wieder wichtige Impulse für das Gelingen der Arbeit. Herr Prof. Dr. Michael Dellnitz legte bereits während meines Studiums wichtige mathematische Grundlagen, die auch für die vorliegende Arbeit sehr hilfreich waren. Umso mehr freut mich, dass er das Korreferat übernommen hat und für beides danke ich ihm. Ich bedanke mich auch bei Herrn Prof. Dr.-Ing. Eugeny Kenig und Prof. Dr.-Ing. Detmar Zimmer für ihr Mitwirken in der Promotionskommission.

Auch den Mitgliedern meiner „Promotionsrunde“, Julia Timmermann, Peter Reinold und Alexander Löffler danke ich sehr herzlich für viele Stunden angeregter Diskussion, unzählige Hinweise fachlicher und weniger fachlicher Art sowie das sorgfältige Korrekturlesen der schriftlichen Arbeit. Ganz besonders bedanke ich mich bei Julia, mit der ich über Jahre hinweg ein Büro geteilt habe, für die harmonische, fruchtbare Zusammenarbeit und vielfältige Unterstützung. Dank gebührt auch meinen SFB-Kollegen Kathrin Flaßkamp, Katrin Witting, Christian Hölscher und Jan Henning Kefler, mit denen ich gerne zusammengearbeitet und publiziert habe. Ich möchte mich auch bei meinen ehemaligen studentischen Hilfskräften Tanja Schmüdderrich und Simon Olma bedanken für die Unterstützung bei der Modellierung und Aufbereitung der Ergebnisse.

Zwei weitere Personen haben entscheidend zum erfolgreichen Entstehen dieser Arbeit beigetragen. Zum einen ist dies meine Frau Silke, die mir immer den Rücken frei gehalten hat, mich stets motiviert, bestärkt und begleitet hat. Zudem ist auch meine Mutter Angelika zu nennen, die mir seit meiner Kindheit über die gesamte Ausbildungszeit hinweg liebevoll zur Seite stand, mich gefördert, beraten und auf vielfältigste Weise unterstützt hat. Beiden danke ich von ganzem Herzen.

Altenbeken, im März 2014

Martin Krüger

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation und Zielsetzung	1
1.2	Selbstoptimierende mechatronische Systeme	3
1.2.1	Grundlagen und Definition der Selbstoptimierung	3
1.2.2	Strukturierung und hierarchisches Modell selbstoptimie- render Systeme	6
1.3	Anwendungsbeispiel Feder-Neige-Prüfstand	11
1.4	Aufbau der Arbeit	15
2	Modellordnungsreduktion durch Interpolation der Übertragungs- funktion	17
2.1	Überblick und historische Entwicklung	17
2.2	Ausgangspunkt, Reduktionsansatz und Bewertungskriterien	20
2.3	Manuelle rationale Interpolation	24
2.3.1	Impliziter Momentenabgleich	25
2.3.2	Arnoldi-Algorithmus	27
2.4	H_2 -optimale tangentielle Interpolation	29
2.4.1	Tangentiale Interpolation	30
2.4.2	Theoretische Grundlagen der \mathcal{H}_2 -optimalen Interpolation	32
2.4.3	Iterative Berechnung des reduzierten Modells	34
2.5	Anwendung am Beispiel des Aktormoduls	37
3	Parametrische Modellordnungsreduktion	45
3.1	Manuelle rationale Interpolation parametrischer Systeme	45
3.1.1	Theoretische Grundlagen	46
3.1.2	Rekursiver objektorientierter Arnoldi-Algorithmus	51
3.2	Matrix Interpolation	58
4	Mehrzieloptimierung selbstoptimierender Systeme	69
4.1	Grundlagen der Mehrzieloptimierung	69
4.2	Hierarchische Mehrzieloptimierung	75
4.3	Optimierungsmodell für selbstoptimierende Systeme	78
4.4	Anwendungen der Mehrzieloptimierung	81
4.4.1	Mehrzieloptimierung des Aktormoduls	82
4.4.2	Mehrzieloptimierung des Feder-Neige-Prüfstands	86

5	Parametrische Modellordnungsreduktion paretooptimaler Systeme	91
5.1	Parametrisierung von Paretomengen	93
5.2	Interpolation paretooptimaler Systeme	97
5.3	Parametrische Modellordnungsreduktion paretooptimaler Systeme	104
5.4	Resultate am Feder-Neige-Prüfstand	108
5.4.1	PMOR des Aktormoduls des Feder-Neige-Prüfstands	109
5.4.2	PMOR des gesamten Feder-Neige-Prüfstands	110
5.5	Erweiterungsmöglichkeiten	115
5.5.1	Behandlung nichtlinearer Systeme	116
5.5.2	Interpolation höherer Ordnung	117
5.5.3	Behandlung von drei oder mehr Zielfunktionen	119
6	Anwendungsgebiete parametrischer reduzierter Modelle	121
6.1	Hierarchische Optimierung des Feder-Neige-Prüfstands	121
6.2	Hierarchische Optimierung eines vernetzten Prüfstands	124
6.2.1	Intelligentes Antriebsmodul	124
6.2.2	Vernetzter Prüfstand	126
6.2.3	Optimierungsergebnisse	128
6.3	Zielfunktionsbasierte Regelung selbstoptimierender Systeme	130
7	Resümee und Perspektiven	135
8	Literaturverzeichnis	137

Abkürzungsverzeichnis

AMS	autonomes mechatronisches System
FNPF	Feder-Neige-Prüfstand
GAIO	Global Analysis of Invariant Objects
GFK	glasfaserverstärkter Kunststoff
iAM	intelligentes Antriebsmodul
IRKA	Iterative Rational Krylov Algorithm
MFM	mechatronisches Funktionsmodul
MOP	Mehrzieloptimierungsproblem
MOR	Modellordnungsreduktion
NBI	Normal Boundary Intersection
OCM	Operator-Controller-Modul
PMOR	parametrische Modellordnungsreduktion
POD	Proper Orthogonal Decomposition
SVD	Singular Value Decomposition
VMS	vernetztes mechatronisches System

Symbolverzeichnis

A_i	Systemmatrix zu α_i
ΔA_i	Steigung der linearen Interpolation paretooptimaler Systeme
α	Parametrierung der Paretomenge
α_i	Stützstellen für die Interpolation paretooptimaler Systeme
$\bar{\alpha}_i$	Parametrierung relativ zur Stützstellenverteilung
\tilde{b}_i, \tilde{c}_i	Tangentialvektoren der tangentialen Interpolation
\mathcal{D}	Definitionsbereich eines MOP
Δ	Kriterium zur Abschätzung des Interpolationsfehlers der paretooptimalen Systeme
e	Reduktionsfehler, d. h. Ausgang des Fehlersystems
E, A, B, C, D	Matrizen der Zustandsdarstellung eines linearen Systems
E_r, A_r, B_r, C_r, D_r	Matrizen der Zustandsdarstellung des reduzierten Systems
ε_D	Deflationsschranke Arnoldi-Algorithmus
ε_I	Abbruchschranke IRKA
ε_S	Abbruchschranke für Stützstellenverteilung
$F(p)$	Zielfunktionsvektor
$f_i(p)$	Zielfunktion
G	Übertragungsfunktion
G_r	Übertragungsfunktion des reduzierten Systems
I_n	Einheitsmatrix der Dimension n
$\mathcal{K}(\cdot, \cdot)$	(Block-)Krylov-Unterraum
$\mathcal{K}_q(\cdot, \cdot)$	(Block-)Krylov-Unterraum der Ordnung q
Λ	Diagonalmatrix der Eigenwerte
λ_i	Eigenwert
$M_k(s_0)$	Moment der Ordnung k der Übertragungsfunktion
$M_{r,k}(s_0)$	Moment der Ordnung k der Übertragungsfunktion des reduzierten Systems
$\text{mpp}(\alpha)$	stückweise lineare Interpolation der Systemmatrix $A(s(\alpha))$
M_i, T_i	Transformationsmatrizen der Matrix Interpolation
μ_i, σ_i	Interpolations- bzw. Entwicklungspunkte der tangentialen Interpolation

n_o	Anzahl Zielfunktionen
n_p	Anzahl System- / Optimierungsparameter
n_S	Anzahl Stützstellen / Diskretisierungspunkte
n_u	Anzahl Systemeingänge
n_x	Dimension des Zustandsraums
n_y	Anzahl Systemausgänge
ω_i	Gewichtsfunktion für Matrix Interpolation
p	Vektor der Systemparameter
P_F	Paretomenge eines MOP mit Zielfunktionsvektor F
Ψ	Rechtseigenvektoren
p^*	paretooptimaler Punkt
q	Ordnung des reduzierten Modells
R	Matrix für Projektion auf gemeinsamen Unterraum
s_0, s_i	Entwicklungspunkt für Momentenabgleich
$s(\alpha)$	Parametrierungsfunktion
u	Eingangsvektor
u_r	Eingangsvektor des reduzierten Systems
Υ	Linkseigenvektoren
V, W	Projektionsmatrizen der Modellordnungsreduktion
x	Zustandsvektor
x_r	Zustandsvektor des reduzierten Systems
y	Ausgangsvektor
y_r	Ausgangsvektor des reduzierten Systems

1 Einleitung

Diese Arbeit ist im Rahmen des Sonderforschungsbereichs 614 – Selbstoptimierende Systeme des Maschinenbaus – entstanden [sfb13]. Einer der wesentlichen Unterschiede eines selbstoptimierenden Systems im Vergleich zu einem gewöhnlichen mechatronischen System besteht darin, dass alle zu erzielenden Eigenschaften des Systems in Form von Zielen ausgedrückt werden. Dies betrifft sowohl die für den Betrieb notwendigen Eigenschaften wie Stabilität oder Verlässlichkeit, als auch wünschenswerte Eigenschaften wie etwa ein geringer Energieverbrauch oder bei Fahrzeugen der komfortable Transport von Personen oder Gütern.

Die Grundidee der Selbstoptimierung besteht in der selbstständigen Anpassung der zu verfolgenden Ziele während des Betriebs an veränderliche Umfedeinflüsse. Eine daraus resultierende zielkonforme Anpassung des Systemverhaltens wird vom selbstoptimierenden System autonom vorgenommen. Zur Bestimmung optimaler Systemkonfigurationen spielen Mehrzielloptimierungsverfahren eine entscheidende Rolle. Insbesondere die modellbasierte Mehrzielloptimierung, bei der basierend auf Modellen des dynamischen Verhaltens Optimierungsprobleme definiert und gelöst werden, hat sich als Mittel der Wahl erwiesen, um selbstoptimierende Systeme auslegen zu können.

1.1 Motivation und Zielsetzung

Selbstoptimierende Systeme verfügen über eine hohe Komplexität. Sie bestehen aus vielen verschiedenen Modulen und Teilsystemen, die sich oftmals selbst in gewisser Weise „intelligent“ verhalten. Eine modellbasierte Entwicklung selbstoptimierender Systeme führt daher unweigerlich zu der Herausforderung, mit umfangreichen, komplexen Modellen umgehen zu können. Insbesondere Simulationen des Systemverhaltens, die für eine modellbasierte Entwicklung unerlässlich sind, erfordern sehr oft lange Berechnungszeiten.

Die im Rahmen selbstoptimierender Systeme auftretenden Mehrzielloptimierungsprobleme sind oft ebenfalls komplex und können nur mit hohem Rechenaufwand gelöst werden, vor allem wenn eine große Zahl von Zielen und Optimierungsparametern berücksichtigt werden soll. Basiert die Mehrzielloptimierung zusätzlich auf Simulationen komplexer Modelle, was oftmals unvermeidbar ist, so verstärkt sich das Problem der Berechenbarkeit. Die Lösung des Mehrzielloptimierungsproblems, d. h. die Bestimmung optimaler Systemkonfigurationen des selbstoptimierenden Systems wird extrem zeitaufwändig.

Eine Reihe von Vorarbeiten, die nachfolgend beschrieben sind, haben sich bereits mit dieser Problematik beschäftigt und dienen als Ausgangspunkt für die vorliegende Arbeit. Für einen korrekten und sicheren Betrieb des Gesamtsystems müssen zum einen die Teilsysteme entsprechend ausgelegt sein, zum anderen muss aber auch das Zusammenspiel koordiniert werden. Daher handelt es sich bei selbstoptimierenden Systemen um verteilte Systeme, die sowohl informativ-technisch als auch physikalisch miteinander verkoppelt sind.

Zur Entwicklung verteilter selbstoptimierender Systeme sind Strukturierungsverfahren und spezielle Architekturen innerhalb des Sonderforschungsbereichs entstanden [LHLH01, HO04, Hes06]. Die Strukturierung führt zu einer hierarchischen Anordnung der Teilsysteme, auf deren Basis ein Ansatz zur hierarchischen Modellierung entwickelt wurde [MAK⁺08]. Mit dessen Hilfe können geeignete mathematische Modelle des dynamischen Verhaltens der Hierarchie-Elemente aufgestellt werden. In diesem Kontext fand bereits eine erste Anwendung von Verfahren der Modellordnungsreduktion statt, um die Modellierungstiefe beim Übergang von einer Hierarchieebene auf die nächste systematisch zu verringern.

Auf der Basis dieses hierarchischen Modells entstanden Verfahren zur hierarchischen Optimierung selbstoptimierender Systeme [MAK⁺08, Mün12]. Es handelt sich hierbei um eine verteilte Mehrzieloptimierung, die „bottom-up“ durchgeführt wird. Für jedes Hierarchieelement werden optimale Systemeinstellungen berechnet. Durch diese hierarchische Optimierung wird die Komplexität der zu betrachtenden Mehrzieloptimierungsprobleme verringert. Wird die Optimierung überlagert Hierarchieelemente auf die optimalen Konfigurationen der unterlagerten Elemente eingeschränkt, hat dies eine geringere Anzahl an Optimierungsparametern zur Folge.

Das Ziel dieser Arbeit besteht in der Weiterentwicklung des hierarchischen Ansatzes. Der Schwerpunkt liegt hierbei auf dem Einsatz der Modellordnungsreduktion. Durch die Anpassung bestehender und Entwicklung neuer Methoden soll vor allem eine stärkere Verzahnung von hierarchischer Modellierung und hierarchischer Optimierung erreicht werden. Zur Lösung des hierarchischen Optimierungsproblems werden die zu den unterlagerten Systemen gehörenden Optimierungsparameter vom Optimierungsalgorithmus permanent variiert. Momentan erfordert dies bei einer Verwendung reduzierter Systeme die Durchführung einer Modellordnungsreduktion in jedem Optimierungsschritt. Mit Hilfe der parametrischen Modellordnungsreduktion soll zukünftig auf die Durchführung der Reduktion innerhalb der Optimierung verzichtet werden. Stattdessen soll die Parameterabhängigkeit des reduzierten Modells die für die Optimierung notwendige Variabilität sicherstellen.

Neben einer Effizienzsteigerung der hierarchischen Optimierung ist die parametrische Modellordnungsreduktion auch für die hierarchische Modellierung im all-

gemeinen vorteilhaft. Durch die Modellordnungsreduktion entsteht eine Kapselung der unterlagerten Systeme innerhalb des hierarchischen Modells. Bisher sind die gekapselten Modelle jedoch nachträglich nicht mehr veränderbar, was durch eine explizite Parameterabhängigkeit behoben werden kann. Zudem wird die Reduktion bisher manuell durchgeführt. Eine weitgehend automatisiert ablaufende parametrische Modellordnungsreduktion ist wünschenswert und stellt daher eine weitere Zielsetzung dieser Arbeit dar.

1.2 Selbstoptimierende mechatronische Systeme

Die Grundlagen selbstoptimierender Systeme sind in den beiden folgenden Abschnitten 1.2.1 und 1.2.2 in Form eines kurzen Überblicks dargestellt. Insbesondere das Konzept zur Strukturierung selbstoptimierender Systeme beschreibt den Ausgangspunkt und den Rahmen für die in dieser Arbeit entstandene Methode.

1.2.1 Grundlagen und Definition der Selbstoptimierung

Das charakteristische Merkmal eines selbstoptimierenden Systems ist die Fähigkeit, autonom auf sich verändernde Umweltbedingungen zu reagieren. Es handelt sich daher bei selbstoptimierenden Systemen um handlungsfähige Systeme, die über ein gewisses Maß an inhärenter Intelligenz verfügen. Die Definition der Selbstoptimierung ist in [ADG⁺08] zu finden und lautet:

Unter Selbstoptimierung eines technischen Systems wird die endogene Anpassung der Ziele des Systems auf veränderte Einflüsse und die daraus resultierende zielkonforme autonome Anpassung der Parameter und ggf. der Struktur und somit des Verhaltens dieses Systems verstanden. Damit geht Selbstoptimierung über die bekannten Regel- und Adaptionsstrategien wesentlich hinaus; Selbstoptimierung ermöglicht handlungsfähige Systeme mit inhärenter „Intelligenz“, die in der Lage sind, selbstständig und flexibel auf veränderte Betriebsbedingungen zu reagieren.

Die wesentliche Erweiterung selbstoptimierender Systeme gegenüber klassischen regelungstechnischen Systemen besteht in der Einführung des „Zielgedankens.“ Hiermit ist das Bestreben gemeint, sämtliche Anforderungen sowie die gewünschten und benötigten Systemeigenschaften als Ziele zu formulieren. Dies ermöglicht eine sehr abstrakte, allgemein verständliche Beschreibung des Systems.

Es wird hierbei zwischen drei Klassen von Zielen unterschieden. Die *inhärenten Ziele* sind durch den Zweck und die Funktionsweise des Systems vorgegeben. Beispiele für inhärente Ziele sind die Stabilität des Systems oder die Verlässlichkeit während des Betriebs.

Neben den inhärenten Zielen beschreiben die *externen Ziele* alle Anforderungen, die von außen an das System herangetragen werden. Die externen Ziele sind so vielfältig wie der Einsatzbereich selbstoptimierender Systeme. Sie reichen von dem Wunsch das System energie- oder ressourceneffizient zu betreiben bis hin zu komfortablem oder zeitoptimalem Transport von Personen und Gütern. Veränderbare externe Ziele stellen zudem eine Eingriffsmöglichkeit des Benutzers auf das Systemverhalten zur Laufzeit dar.

Aus der Gesamtmenge der Ziele, d. h. aus der Vereinigung von inhärenten und externen Zielen, werden die *internen Ziele* bestimmt. Diese stellen eine Auswahl der Ziele sowie eine Priorisierung für den momentanen Zeitpunkt dar. Ein selbstoptimierendes System verfolgt zu jedem Zeitpunkt seine internen Ziele, die zum sogenannten *Zielsystem* zusammengefasst werden. Durch eine Veränderung der internen Ziele zur Laufzeit erhält man eine abstrakte, allgemein verständliche Schnittstelle, um das Systemverhalten anpassen zu können. Diese Schnittstelle flexibel und robust zu gestalten, ist eine der Herausforderungen im Entwurf selbstoptimierender Systeme. Insbesondere weil an dieser Schnittstelle eine Verbindung zwischen der abstrakten, leicht vorstellbaren Welt der Ziele und den technischen Eingriffsmöglichkeiten in einem komplexen mechatronischen System geschaffen werden muss. Beispielsweise weist ein Fahrgast das System an, im Folgenden energiesparender zu fahren. Daraufhin muss das System unter Berücksichtigung der aktuellen Situation und der erforderlichen Beschränkungen verschiedene Komponenten verstellen, vielleicht sogar gewisse Funktionen an- oder abschalten. Hierzu ist es unerlässlich, die oftmals komplexen mathematischen Verfahren und Regelstrategien so weit wie möglich zu automatisieren und die Entwurfsfreiheiten sinnvoll, d. h. mit Bezug zu den Zielen, zu abstrahieren. Die in dieser Arbeit entstandene Methode zur parametrischen Modellordnungsreduktion paretooptimaler Systeme trägt auch zu diesem Aspekt bei.

Bereits aus der Definition wird ersichtlich, dass Selbstoptimierung keine Methode darstellt, die einmalig angewendet wird. Vielmehr vollzieht sich Selbstoptimierung durch das wiederholte Ausführen des *Selbstoptimierungsprozesses*. Dieser besteht aus den drei Teilen: *Analyse der Ist-Situation*, *Bestimmung der Systemziele* und *Anpassung des Systemverhaltens*. Während des ersten Schrittes, der Analyse der Ist-Situation, wertet das System sein Umfeld möglichst umfassend aus. Hierzu gehören die Erfassung und Auswertung von Sensorsignalen, die sowohl Informationen über den Systemzustand, als auch über die Umwelt liefern. Weiterhin kommen modellbasierte Beobachter sowie oftmals zusätzliche Methoden zur Erfassung der aktuellen Situation zum Einsatz, die über die Ansätze klassischer Regelungstechnik hinausgehen. Beispielsweise hat sich der Einsatz von Störgrößenbeobachtern in Kombination mit iterativen Lernverfahren sowohl im Kontext schienengebundener Fahrzeuge [TMV06], als auch für Straßenfahrzeuge [BKR⁺11] als geeigne-

te Methodik erwiesen. Sobald mehrere selbstoptimierende Systeme miteinander agieren, können über Kommunikationsnetze weitere Daten zur Bestimmung der Ist-Situation ausgetauscht werden. Ein Beispiel hierfür stellt die Konvoi-Fahrt der nachfolgend noch genauer beschriebenen RailCabs dar [HTS⁺08]. Allgemein ist es von Vorteil, eine möglichst breite und verlässliche Daten- bzw. Wissensbasis für die nachfolgenden Schritte des Selbstoptimierungsprozesses zur Verfügung zu haben.

Basierend auf den Daten des ersten Schrittes wird das aktuell gültige Zielsystem erstellt. Formal unterliegt dieser Schritt keinerlei Restriktionen. Es ist prinzipiell möglich, Ziele aus der Menge der inhärenten und externen Ziele neu aufzunehmen oder gänzlich neue Ziele zu generieren. In vielen Fällen ist jedoch eine geänderte Priorisierung der vorhandenen Ziele ausreichend. Planungsverfahren stellen hierfür eine geeignete Methodik dar, die aktuellen Ziele unter Abschätzung der zukünftigen Einflüsse festzulegen. Neben rein diskreten Planungsverfahren, mit deren Hilfe beispielsweise die Route eines Fahrzeugs innerhalb eines bekannten Verkehrsnetzes bestimmt werden kann [KSWRV09], können auch detailliertere Planungsverfahren, die sogenannte hybride Planung, eingesetzt werden. Diese nutzt Simulationen vereinfachter Modelle, um das zukünftige Verhalten vorherzusagen und auf diese Weise das aktuelle Zielsystem so festzulegen, dass beispielsweise vorgegebene Energieschranken eingehalten werden [MAK⁺08, AEH⁺11]. Eine Anpassung der Priorisierung in Form von Zielgewichtungen lässt sich auch mit regelungstechnischen Methoden erreichen. Der in Kapitel 6 beschriebene zielfunktionsbasierte Regler ist ein Beispiel hierfür.

Der letzte Schritt des Selbstoptimierungsprozesses beschreibt die tatsächliche Änderung des Systemverhaltens. Hier sind alle Methoden zu finden, durch deren Anwendung das Systemverhalten verändert wird. Hierbei kann es sich um eine einfache Parameteranpassung handeln, wenn beispielsweise ein Reglerparameter verändert wird. Auch die Parameter von Modellen, die in modellbasierten Beobachtern eingesetzt werden, können angepasst werden, wenn etwa als Resultat von Identifikationsverfahren aktuellere Parameterwerte vorliegen.

Neben Parameteranpassungen kann auch die Struktur des Systems angepasst werden. Dies ist z. B. der Fall, wenn zwischen verschiedenen Reglern umgeschaltet wird. Ein anschauliches Beispiel hierfür ist die Umschaltung von einem Abstands- auf einen Geschwindigkeitsregler im Kontext von Straßenfahrzeugen. Es ist aber auch möglich, verschiedene Anteile einer komplexen Regelung ein- und auszuschalten. Beispielsweise wird in [VT08] eine Störgrößenkompensation situationsabhängig genutzt oder abgeschaltet. In diesem Beispiel kann eine harte Umschaltung von einer Konfiguration zur anderen stattfinden. In Fällen, in denen dies nicht möglich ist, sind komplexere Umschaltstrategien notwendig. Die flachheitsbasierte Umschaltung [OT08] stellt ein Beispiel hierfür dar. Eine weitere Art

der Strukturanpassung für den Fall eines Sensor- oder Aktorausfalls ist die Rekonfiguration der Regelung [Ste05, BKLS06]. Hierbei wird durch das zusätzliche Einfügen eines Rekonfigurationsblocks in die Regelungsstruktur das fehlerhafte Systemverhalten ausgeglichen.

Es ließen sich noch weitere Anwendungsbeispiele und Ausprägungen des Selbstoptimierungsprozesses anführen. Eine ausführliche Beschreibung verschiedener Umsetzungen ist in [ADG⁺08] zu finden. All diese Anwendungsbeispiele besitzen die Gemeinsamkeit, nur durch komplexe Modelle und eine umfangreiche Informationsverarbeitung realisierbar zu sein. Dies ist ein weiteres Kennzeichen selbstoptimierender Systeme und auch ein Nachteil, da sowohl die Entwickler, als auch die Benutzer diese Komplexität beherrschen müssen. Das kann nur möglich sein, wenn die Informationsverarbeitung in geeigneter Weise strukturiert ist und an die Bedürfnisse der Selbstoptimierung angepasste Architekturen vorhanden sind. Auf derartige Strukturierungsverfahren, die innerhalb des Sonderforschungsbereichs entwickelt wurden, geht der folgende Abschnitt ein.

1.2.2 Strukturierung und hierarchisches Modell selbstoptimierender Systeme

Die vielfältigen Anforderungen und Zielsetzungen an selbstoptimierende Systeme resultieren in einer hohen Komplexität des Systems und vor allem der Informationsverarbeitung. Diese Komplexität wird durch aufwändige Methoden und Verfahren weiter erhöht. Ein systematischer Entwurf selbstoptimierender Systeme ist daher nur möglich, wenn das komplexe Gesamtsystem sinnvoll strukturiert ist.

In der Vergangenheit hat sich eine Strukturierung anhand der Funktionen des Systems als sehr gut geeignet erwiesen. Der Begriff der Funktion stammt aus der Konstruktionslehre und beschreibt dort den „*gewollten Zusammenhang zwischen Eingang und Ausgang eines Systems mit dem Ziel, eine Aufgabe zu erfüllen*“ [PBFG05]. Diese Definition kann für selbstoptimierende Systeme übernommen werden.

Die Funktionen eines Systems lassen sich stets in Teilfunktionen zerlegen und in einer hierarchischen Funktionsstruktur anordnen. Diese wird bereits in den frühen Phasen des Entwurfs selbstoptimierender Systeme aufgestellt und bildet eines der insgesamt acht Partialmodelle der Prinziplösung selbstoptimierender Systeme [GFDK08a, GFDK08b].

Den Elementen der Funktionshierarchie lassen sich bei mechatronischen Systemen spezielle Strukturierungselemente zuordnen, die erstmals in [LHLH01] vorgestellt wurden. Diese Strukturierung wird auch als Makrostruktur eines Systems bezeichnet. Die Basis bildet das sogenannte *mechatronische Funktionsmodul (MFM)*. Es

besteht aus den vier Grundbausteinen mechanisches Grundsystem, Sensorik, Aktorik und Informationsverarbeitung. Nach [Hes06] realisiert ein MFM eine bestimmte Funktion der Funktionshierarchie.

Die MFM repräsentieren jeweils ein eigenes Aggregat des Gesamtsystems und werden oftmals als Baugruppe realisiert. Dabei sind die MFM physikalisch miteinander verkoppelt, durch mechanische Kopplungen, Stoff- und Energieflüsse. Auf der obersten Ebene der Funktionshierarchie befindet sich die Gesamtfunktion des Systems. Das hierzu korrespondierende mechatronische Strukturelement wird als *autonomes mechatronisches System (AMS)* bezeichnet. Es verfügt wie ein MFM über ein mechanisches (physikalisches) Grundsystem, Sensorik und Informationsverarbeitung. Die Aktorik wird jedoch durch die unterlagerten MFM realisiert. Das AMS kann autonom in seiner Umwelt agieren und ist nicht physikalisch mit anderen Systemen verbunden.

Es ist jedoch möglich, mehrere AMS informationstechnisch zu verkoppeln. Die AMS können dann ihr eigenes Verhalten an die übrigen Systeme anpassen und z.B. ein gemeinsames Ziel oder eine übergeordnete Funktion erfüllen. Im Fall solcher informationstechnisch verkoppelter AMS spricht man von einem *vernetzten mechatronischen System (VMS)*, welches das dritte Strukturierungselement darstellt. Ein VMS besteht nicht aus den vier Grundbausteinen. Es benötigt lediglich eine Informationsverarbeitung, die auch ohne eigene Hardware realisiert werden kann, indem die Ressourcen der zugehörigen AMS verwendet werden. Ein im Sonderforschungsbereich 614 intensiv studiertes VMS ist der Konvoi aus mehreren RailCabs. Hier wird beispielsweise in der Informationsverarbeitung auf VMS-Ebene eine Konvoi-Regelung implementiert [HTS⁺08].

Die Strukturierung verringert die Komplexität für jedes einzelne Element im Vergleich zum Gesamtsystem, insbesondere im Bereich der Informationsverarbeitung. Wird nun jedoch in jedem Strukturelement ein Selbstoptimierungsprozess implementiert, so ist eine weitere Strukturierung der Informationsverarbeitung notwendig. Diese Strukturierung, mit deren Hilfe die Komplexität weiter reduziert wird, wird auch als Mikrostruktur bezeichnet [ADG⁺08]. Das zentrale Element ist das *Operator-Controller-Modul (OCM)*[HO04], das aus den drei Schichten besteht, die in Bild 1-1 dargestellt sind.

Auf der untersten Ebene befindet sich der *Controller*. In ihm sind die Elemente der klassischen Regelungstechnik zu finden: ein oder mehrere Regler zur Berechnung der Stellgrößen für die Aktorik sowie einfache Signalverarbeitungsalgorithmen zur Aufbereitung der Sensordaten. Einfache Modelle, enthalten beispielsweise in Zustandsbeobachtern oder für dynamische Vorsteuerungen, sind ebenfalls denkbar. Das entscheidende Kennzeichen ist, dass der Controller die Sensordaten in harter Echtzeit verarbeitet und entsprechende Stellsignale ebenfalls unter harten

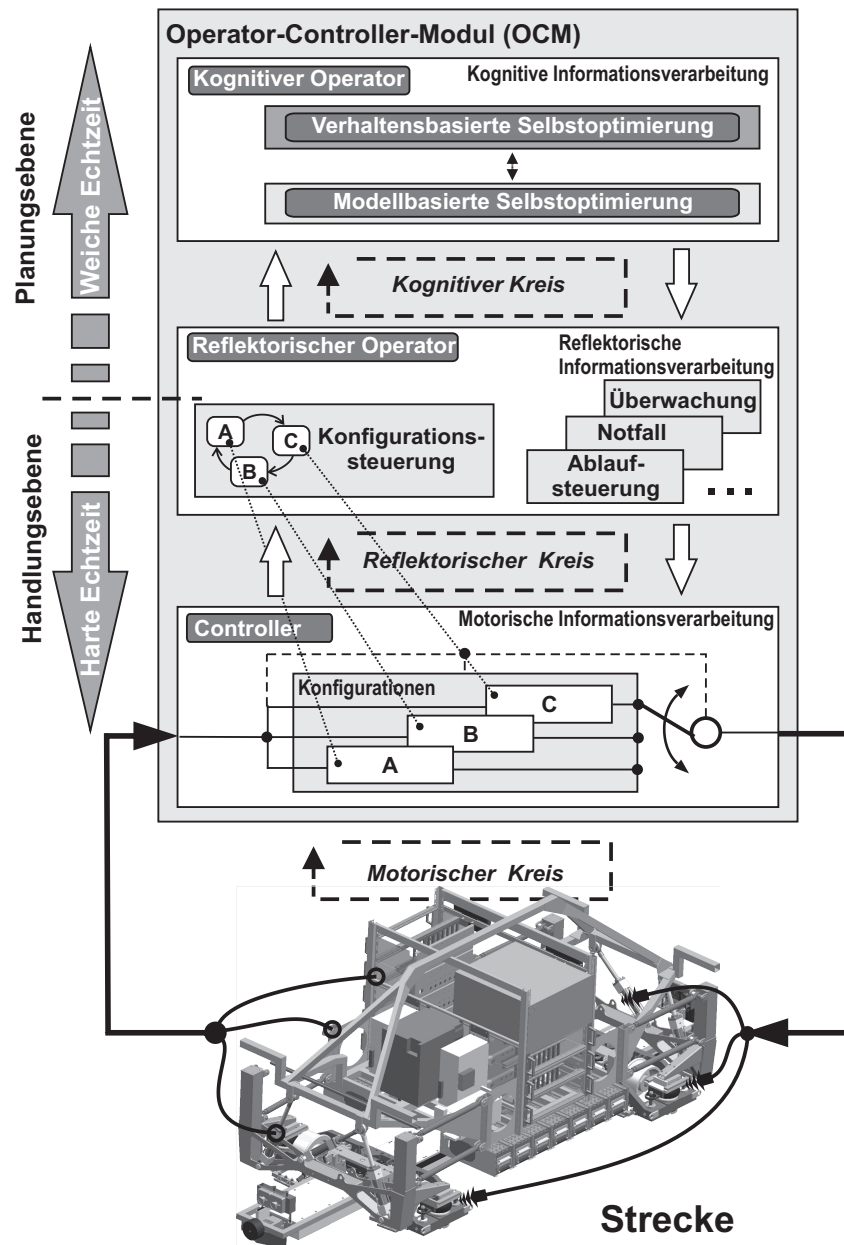


Bild 1-1: Operator-Controller-Modul (OCM) zur Strukturierung der Informationsverarbeitung selbstoptimierender Systeme [HO04]

Echtzeitanforderungen an die Aktorik ausgibt, um ein gewünschtes dynamisches Verhalten des zugehörigen physikalischen Systems sicher zu stellen.

Der *Reflektorische Operator* bildet die mittlere Schicht des OCM. Er hat keinen direkten Zugriff auf die Aktorik der Regelstrecke, sondern beeinflusst das Systemverhalten, indem er Änderungen am Controller vornimmt. Diese Änderungen beinhalten sowohl Parameter-, als auch strukturelle Änderungen. Sie können einen Umschaltvorgang zwischen verschiedenen Reglern darstellen oder auch das Hinzufügen oder Entfernen von Controller-Komponenten. Wie bereits erwähnt, kann beispielsweise eine Vorsteuerung oder Störgrößenkompensation hinzugenommen oder abgeschaltet werden [VT08].

Die Anpassungen des Controllers können aus Adaptionstrategien oder einer vorgegebenen Ablaufsteuerung resultieren. Ebenso sind im Reflektorischen Operator Sicherheits- und Notfallroutinen hinterlegt. Zur Umsetzung werden oftmals Zustandsautomaten mit einfachen Umschaltbedingungen oder auch mit zeitbehafteten Umschaltprozessen verwendet. Insbesondere im Bereich von Sicherheitsmaßnahmen können Modelle in der Modellierungssprache MechatronicUML verwendet werden. Derartige Modelle besitzen den Vorteil, dass eine formale Verifikation möglich ist, siehe z. B. [HPB12, Pri09].

Da der Reflektorische Operator den Controller konfiguriert, wird er ebenfalls unter harten Echtzeitbedingungen ausgeführt. Er beinhaltet aber zusätzlich die Schnittstellen zu den kognitiven Funktionen, die oftmals nicht echtzeitfähig sind. Die ereignisgesteuerte Umsetzung von Vorgaben aus den kognitiven Funktionen muss mit geeigneten Routinen im Reflektorischen Operator erfolgen. Zudem werden die Sensorsignale bei Bedarf im Reflektorischen Operator zwischengespeichert und in geeigneter Form an die überlagerte Ebene weitergegeben.

Alle Methoden und Verfahren des Selbstoptimierungsprozesses, die nicht in harter Echtzeit ausführbar sind, werden auf der obersten Ebene des OCM im *Kognitiven Operator* implementiert. Die eingesetzten Methoden dienen dazu, Wissen aufzubauen und zur Verbesserung des eigenen Verhaltens zu nutzen. Einen großen Bereich stellen hier modellbasierte Verfahren dar. Die modellbasierte Optimierung nutzt spezielle Modelle des zugehörigen Systems und seiner Umwelt, um optimale Systemkonfigurationen zu berechnen. Werden zukünftige Einflüsse wie etwa gelernte Anregungsprofile für die Optimierung genutzt, ermöglicht dies eine vorausschauende Anpassung des Systemverhaltens. Neben Optimierungsverfahren können an dieser Stelle auch Planungswerkzeuge zur Abschätzung und Anpassung des zukünftigen Verhaltens Verwendung finden.

Die meisten der im Kognitiven Operator eingesetzten Verfahren sind zu rechenintensiv, um in harter Echtzeit ausgeführt zu werden. Dies liegt unter anderem auch an den komplexen Modellen, die in den modellorientierten Verfahren verwendet werden. Eine zeitliche Entkopplung von der harten Echtzeit wird durch

den Reflektorischen Operator möglich. Jedoch ist zu beachten, dass die Ergebnisse innerhalb gewisser Zeitschranken vorliegen müssen, da sonst keine selbstständige Anpassung des Systemverhaltens möglich wird. Daher spricht man an dieser Stelle von weicher Echtzeit, der der Kognitive Operator unterliegt.

Die OCM-Struktur kann in vielfältiger Weise realisiert werden. Je nach betrachtetem System und gegebenen Anforderungen werden einzelne Aspekte ausgeprägter vorhanden sein und andere wiederum vernachlässigt werden. Für eine Reihe von Beispielen sei erneut auf [ADG⁺08] verwiesen.

Die Einführung der Makrostruktur wie auch der Mikrostruktur dienen der Verringerung der Komplexität selbstoptimierender Systeme. Sie strukturieren unterschiedliche Aspekte des Systems und können miteinander kombiniert werden. Wird in jedem Strukturelement der Makrostruktur (MFM, AMS, VMS) die Informationsverarbeitung durch die OCM-Architektur dargestellt, so ergibt sich eine OCM-Hierarchie, die weiterhin die Funktionsstruktur des Systems widerspiegelt.

Ausgehend von der OCM-Hierarchie wird in [MAK⁺08] und [Mün12] ein hierarchisches Modell vorgestellt, das Teil einer dem Kognitiven Operator zuzuordnenden Wissensbasis ist und für modellbasierte Verfahren genutzt werden kann. Die Grundidee dieses hierarchischen Modells besteht darin, auf einer bestimmten Hierarchieebene ein detailliertes Modell des dynamischen Verhaltens des zugehörigen Elements zu erstellen. Dieses Modell beinhaltet sowohl die Dynamik der Regelstrecke, als auch den Regler selbst. Das Verhalten der unterlagerten Systeme muss dabei berücksichtigt werden, da abgesehen von der obersten Hierarchieebene (VMS) die unterlagerten Systeme physikalisch mit dem betrachteten System verkoppelt sind. Um die Komplexität des Modells zu verringern, werden die unterlagerten Systeme jedoch nur in vereinfachter Form eingebunden. Die Vereinfachung erfolgt durch die Anwendung von Verfahren der Modellordnungsreduktion, nachdem ggf. eine Linearisierung durchgeführt wurde. Dieses Prinzip ist in Bild 1-2 dargestellt.

Das Konzept des hierarchischen Modells wird in dieser Arbeit beibehalten. Aufbauend hierauf wird die parametrische Modellordnungsreduktion zur Vereinfachung der unterlagerten Systeme untersucht. Generell besteht der Vorteil der Reduktion der unterlagerten Systeme darin, dass diese in gekapselter Form, nämlich als Zustandsdarstellung, zwischen den Hierarchieelementen ausgetauscht werden können. Auf diese Weise können sie sehr einfach im Modell des überlagerten Systems eingebunden werden. Ein Problem entsteht, wenn das Verhalten der unterlagerten Systeme variabel bleiben soll. Dies ist beispielsweise der Fall, wenn das hierarchische Modell im Rahmen der hierarchischen Optimierung verwendet wird. In den bisherigen Arbeiten wurde diese Problematik dadurch gelöst, dass bei jeder Verhaltensänderung das Reduktionsverfahren erneut angewendet wird.

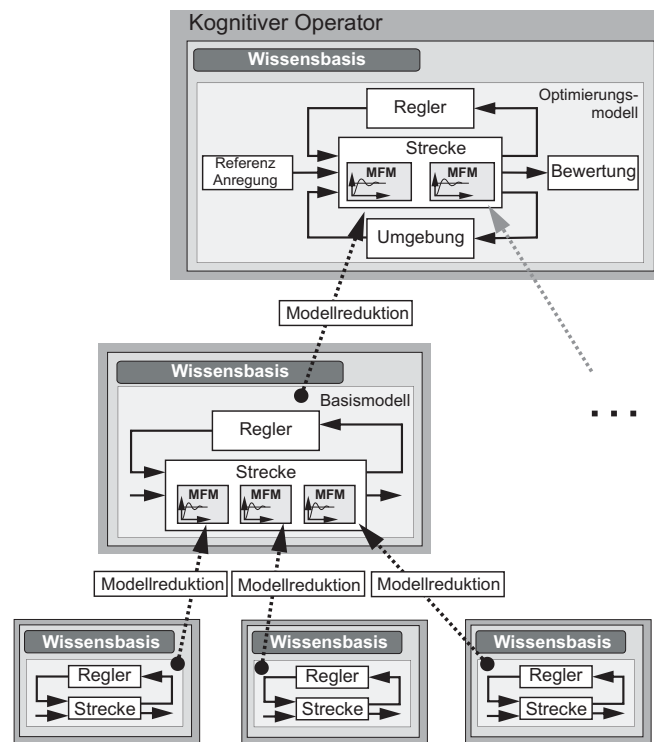


Bild 1-2: Hierarchisches Modell als Teil der Wissensbasis im Kognitiven Operator

Durch den Einsatz der parametrischen Modellordnungsreduktion ist das nicht mehr notwendig. Das parametrische reduzierte Modell kapselt das dynamische Verhalten einschließlich einer gewissen Variabilität in Form der Parameterabhängigkeit. Eine für selbstoptimierende Systeme besonders geeignete Darstellung der Variabilität durch eine spezielle Parametrierung und Interpolation stellt eines der Hauptergebnisse dieser Arbeit dar, siehe hierzu Kapitel 5. Weitergehende Informationen zum hierarchischen Modell, insbesondere welche Aspekte bei der Modellierung zu beachten sind, sind ausführlich in [Mün12] dargestellt. Konkrete Beispiele hierarchischer Modelle befinden sich im nachfolgenden Abschnitt 1.3, in dem das Hauptanwendungsbeispiel dieser Arbeit vorgestellt wird, sowie in Abschnitt 6.2.

1.3 Anwendungsbeispiel Feder-Neige-Prüfstand

Zur Validierung der im Rahmen dieser Arbeit entstandenen Methode werden verschiedene Anwendungsgebiete betrachtet. Zentral sind hierbei Modelle eines Feder-Neige-Prüfstands für die aktive Federung sowie die Neigetechnik des Schienenfahrzeugs RailCab. Sowohl das RailCab-System, als auch der Feder-Neige-

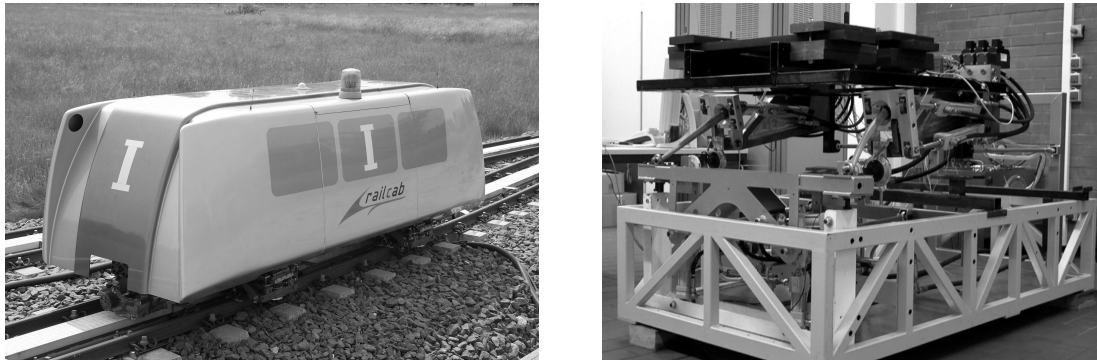


Bild 1-3: links: *RailCab Versuchsfahrzeug* rechts: *Feder-Neige-Prüfstand*

Prüfstand mit der in dieser Arbeit gewählten hierarchischen Modellierung werden im Folgenden vorgestellt.

Die RailCabs bilden den Kern der Neuen Bahntechnik Paderborn. Dabei handelt es sich um ein an der Universität Paderborn entwickeltes, innovatives Verkehrskonzept, siehe [Trä06, NBP13]. Es vereint die Vorteile des Individualverkehrs mit denen des Schienenverkehrs, indem die RailCabs autonom und fahrerlos auf den bestehenden Trassen verkehren. Dadurch wird eine bedarfsgerechte, individuelle Beförderung von Personen und Gütern ermöglicht. Durch den automatisierten Fahrbetrieb können die RailCabs zielrein, d. h. ohne Zwischenhalte, ihr Ziel erreichen, was zu einer deutlichen Verkürzung der Fahrzeiten gegenüber dem konventionellen Schienenverkehr führt. Eine selbstständige Konvoibildung der RailCabs führt zudem zu einem energieeffizienten Betrieb [HTS⁺08].

Jedes RailCab verfügt über mehrere aktive Komponenten, die entsprechend der Ausführungen des vorhergehenden Abschnitts funktionsorientiert zu Modulen zusammengefasst werden. Der Antrieb erfolgt berührungslos durch einen doppelt gespeisten Linearmotor. Analog zum Transrapid [Thy13] befindet sich der Stator in der Schiene und der Läufer im Fahrzeug. Weiterhin besitzt das RailCab ein Spurführungsmodul, mit dem eine aktive Spurführung realisiert wird. Diese ermöglicht einerseits eine komfortable Fahrt, u. a. durch eine Verringerung der Spurkranzanläufe, andererseits den Einsatz passiver Weichen, die für die Konvoibildung unerlässlich sind. Der Fahrkomfort wird zudem durch den Einsatz einer aktiven Federung erhöht. Diese dient darüber hinaus einer aktiven Verstellung der Seitenneigung während Kurvenfahrten.

Seit 2003 ist an der Universität Paderborn eine Versuchsstrecke mit einer Schienlänge von 600m in Betrieb. Zwei RailCab-Versuchsfahrzeuge im Maßstab 1:2.5, siehe Bild 1-3 links, stehen zur Validierung der Technologie zur Verfügung. Darüber hinaus existieren Prüfstände für einzelne Module des RailCabs. Der Feder-Neige-Prüfstand, dessen Modelle in dieser Arbeit als Anwendungsbeispiel

spiele für die Modellordnungsreduktion dienen, stellt einen dieser Prüfstände dar, siehe Bild 1-3 rechts. In Kapitel 6 wird darüber hinaus ein Prüfstand des intelligenten Antriebsmoduls verwendet, mit dem der Luftspalt zwischen Stator und Läufer im RailCab aktiv verändert werden kann. Weitere Details zu diesem Prüfstand sind in Kapitel 6 zu finden.

Mit dem Feder-Neige-Prüfstand können drei Freiheitsgrade des RailCabs untersucht werden: die Hub- und die Wankbewegung sowie die Querbewegung der Aufbaumasse. Der Prüfstand ist als Halbfahrzeugprüfstand realisiert und bildet die vordere bzw. hintere Hälfte des Fahrzeugs nach. Neben einer Aufbaumasse besteht er aus zwei Aktormodulen und einer Anregungseinheit. Im Gegensatz zu der aktiven Federung des Versuchsfahrzeugs sind die Aktormodule geeignet, in Unterflurbauweise ins RailCab eingebaut zu werden, wodurch bei einer Realisierung Bauraum gespart würde. Der Maßstab des Prüfstands beträgt wie beim Versuchsfahrzeug 1:2.5.

Jedes Aktormodul besteht aus drei waagrecht verbauten Hydraulikzylindern. Diese sind an der einen Seite fest mit der Aufbaumasse verbunden. Die Zylinderkolben an der anderen Seite sind an einer Umlenkinematik befestigt, an der auch eine Feder aus glasfaserverstärktem Kunststoff (GFK-Feder) angebracht ist. Durch das Ein- und Ausfahren der Hydraulikzylinder kann auf diese Weise eine Verschiebung des Federfußpunktes erzielt werden. Die andere Seite der GFK-Feder ist mit der Anregungseinheit verbunden. Diese dient der Nachbildung von Störungen in vertikaler und Fahrzeugquerrichtung, die in der Realität auf Gleislagefehler zurückzuführen sind. Die Bewegung der Anregungseinheit wird mit Hilfe von weiteren drei Hydraulikzylindern realisiert. Der Prüfstand wurde im Rahmen der Arbeit [Sch06] entwickelt, der weitere Details zum Aufbau und zu den Komponenten entnommen werden können. Die implementierte Regelung zur Umsetzung der aktiven Federung ist zudem in [VT08] beschrieben.

Eine funktionsorientierte Zerlegung des Feder-Neige-Prüfstands und ein hierarchisches Modell, wie sie in Abschnitt 1.2.2 beschrieben sind, werden in [MAK⁺08] vorgestellt. Die Zerlegung besteht aus einer Hierarchie mit drei Ebenen und insgesamt zehn verschiedenen Elementen. Weitere Details zur Modellierung sind zudem in [Mün12] zu finden.

In dieser Arbeit wird eine hierarchische Zerlegung verwendet, die auf der bestehenden Hierarchie basiert, allerdings nur aus den beiden oberen Ebenen besteht. Wie Bild 1-4 zeigt, wird die untere Ebene von den beiden Aktormodulen gebildet. Die dritte Ebene, die aus den sechs einzelnen Hydraulikzylindern besteht, wird nicht betrachtet, da eine Modellordnungsreduktion der Hydraulikzylinder aufgrund ihrer geringen Systemordnung nicht notwendig ist und die in dieser Arbeit betrachteten Verfahren nicht sinnvoll eingesetzt werden können.

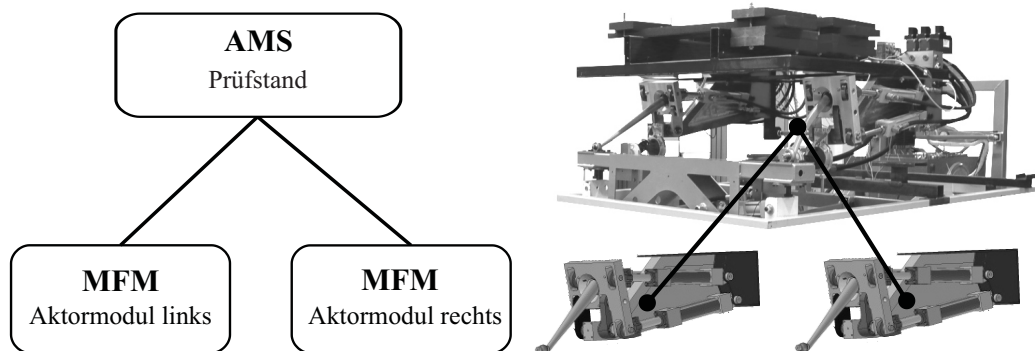


Bild 1-4: Hierarchische Zerlegung des Feder-Neige-Prüfstands mit Angabe der Strukturelemente

Die obere Hierarchieebene wird von dem gesamten Prüfstand gebildet. Da keine weiteren physikalischen Kopplungen zu anderen System existieren, handelt es sich bei dem Prüfstand um ein AMS.

Im Rahmen der hierarchischen Modellierung werden zunächst die Aktormodule separat betrachtet. Die Schnittstellen sind dabei wie in Bild 1-5 dargestellt gewählt. Die Eingangsgrößen werden von den drei Sollpositionen zyl_{soll} der Hydraulikzylinder gebildet. Diese werden im Gesamtsystem aus den Sollvorgaben für die in den GFK-Federn wirkenden Kräfte mit Hilfe einer inversen Kinematik berechnet. Das Modell des Aktormoduls beinhaltet die Dynamik sowie die lokale Regelung der Hydraulikzylinder. Als Ausgänge werden die Position p_{GFK} und die Geschwindigkeit v_{GFK} des Federfußpunktes der GFK-Feder benötigt, um im Gesamtsystem die wirkenden Kräfte zwischen Aufbaumasse und Anregungseinheit berechnen zu können. Hierbei sind die Komponenten in y- und z-Richtung, d. h. horizontal und vertikal, ausreichend, da Bewegungen der Aufbaumasse und der Anregungseinheit in x-Richtung am Prüfstand konstruktiv gesperrt sind. Für die hierarchische Optimierung des Feder-Neige-Prüfstands, auf die in Abschnitt

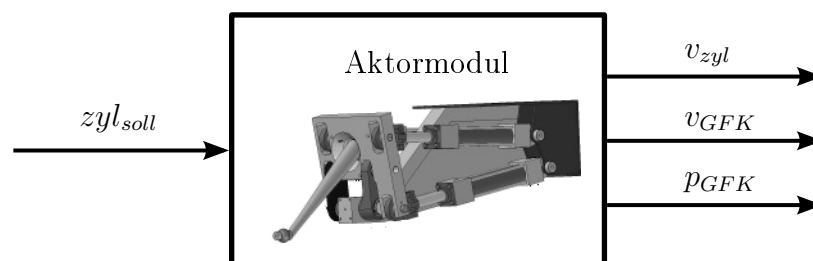


Bild 1-5: Ein- und Ausgänge des Aktormoduls

6.1 eingegangen wird, ist als zusätzlicher Ausgang die Geschwindigkeit v_{zyl} der Hydraulikzylinder erforderlich.

Das Hierarchische Modell des Gesamtsystems beinhaltet die beiden Aktormodule in linearisierter und ggf. reduzierter Form. Zudem enthält es die Dynamik der Anregungseinheit und der Aufbaumasse sowie die Regelung des Prüfstands.

1.4 Aufbau der Arbeit

Kapitel 2 enthält eine generelle Einführung in die Modellordnungsreduktion und eine kurze Übersicht über die historische Entwicklung. Des Weiteren werden die beiden für diese Arbeit relevanten Verfahren der nichtparametrischen Modellordnungsreduktion, die manuelle rationale Interpolation sowie die \mathcal{H}_2 -optimale tangentielle Interpolation, dargestellt.

Aufbauend auf den Grundlagen beschreibt Kapitel 3 die in dieser Arbeit verwendete parametrische Modellordnungsreduktion. Dies ist zum einen die manuelle rationale Interpolation parametrischer Systeme, die mittels eines im Rahmen dieser Arbeit entstandenen objektorientierten Arnoldi-Algorithmus umgesetzt wird. Zum anderen wird die Matrix Interpolation als alternatives Verfahren zur parametrischem Modellordnungsreduktion vorgestellt.

Kapitel 4 widmet sich der Mehrzieloptimierung. Ausgehend von einigen Grundlagen und einer kurzen Übersicht über die verwendeten numerischen Verfahren, wird auf die hierarchische Optimierung sowie auf die Anwendung von Mehrzieloptimierungsverfahren auf selbstoptimierende Systeme eingegangen. Die am Ende dieses Kapitels präsentierten numerischen Ergebnisse dienen als Grundlage für die beiden folgenden Kapitel.

Der Kern der vorliegenden Arbeit, die parametrische Modellordnungsreduktion paretooptimaler Systeme, wird in Kapitel 5 vorgestellt. Basierend auf den in den vorhergehenden Kapiteln dargestellten Grundlagen wird eine neuartige Interpolation der Paretomenge sowie der zugehörigen dynamischen Systeme vorgestellt, die zu einer speziellen, für die parametrische Modellordnungsreduktion besonders geeigneten Systemdarstellung führt. Anhand von zwei Anwendungsbeispielen aus dem Kontext des Feder-Neige-Prüfstands wird die Methode verifiziert.

Zwei Anwendungsgebiete für parametrische reduzierte Systeme werden in Kapitel 6 erläutert. Dies ist zum einen die hierarchische Optimierung, auf die in zwei verschiedenen Anwendungsbeispielen eingegangen wird. Zum anderen wird der Einsatz des reduzierten Modells des Feder-Neige-Prüfstands im Rahmen der Auslegung einer zielfunktionsbasierten Regelung vorgestellt.

Die Arbeit schließt mit einer kurzen Zusammenfassung und einigen Perspektiven für zukünftige Arbeiten in Kapitel 7.

2 Modellordnungsreduktion durch Interpolation der Übertragungsfunktion

Dieses Kapitel gibt einen kurzen Überblick über die Entwicklung der Modellordnungsreduktion (MOR) und den aktuellen Stand der Forschung. Es dient dazu, eine Einführung in die für die vorliegende Arbeit relevanten Verfahren zu geben. Die Darstellung beschränkt sich auf die nicht-parametrische MOR. Die Erweiterungen auf parameterabhängige Systeme basieren auf den hier vorgestellten Verfahren, werden jedoch separat in Kapitel 3 dargestellt.

Der erste der insgesamt fünf Abschnitte dieses Kapitels beschreibt die historische Entwicklung und ordnet die im Rahmen dieser Arbeit verwendeten MOR-Verfahren in den Stand der Forschung ein. Nachfolgend werden in Abschnitt 2.2 unabhängig von konkreten Verfahren die Grundlagen der MOR hergeleitet. Anschließend werden mit der manuellen rationalen Interpolation in Abschnitt 2.3 und der \mathcal{H}_2 -optimalen Interpolation in Abschnitt 2.4 die beiden in dieser Arbeit verwendeten Modellreduktionsverfahren eingeführt. Ein erstes Anwendungsbeispiel wird im letzten Abschnitt 2.5 zur Gegenüberstellung der Verfahren verwendet.

2.1 Überblick und historische Entwicklung

Verfahren zur Modellordnungsreduktion werden eingesetzt, um eine komplexe mathematische Beschreibung eines dynamischen Systems durch eine einfachere Beschreibung zu ersetzen. Unter Komplexität ist in diesem Kontext die Anzahl der Systemgleichungen zu verstehen. Insbesondere für lineare Systeme, die noch immer das Hauptanwendungsgebiet der Modellordnungsreduktion darstellen, trägt die Systemordnung wesentlich zur Komplexität bei. Im Folgenden wird ein kurzer Überblick über die Entstehung der Modellordnungsreduktion gegeben mit dem Ziel, die in dieser Arbeit verwendeten Verfahren einordnen zu können. Als Grundlage hierfür dienten die Einführungen in das Themengebiet aus [Ant05, Kub08, SVR08].

Erste Ansätze, die eine mathematische Funktion durch eine einfachere Funktion ersetzen, sind wesentlich älter als das Forschungsgebiet der MOR. Beispielsweise beschäftigte sich Fourier bereits 1807 mit der Annäherung einer komplizierten Funktion durch trigonometrische Funktionen [Fou07]. Die auch im Kontext der Modellordnungsreduktion verwendete Padé-Approximation aus dem Jahre 1892 ist ein weiteres Beispiel für die Approximation einer mathematischen Funktion, in diesem Fall durch eine rationale Funktion [Pad92].

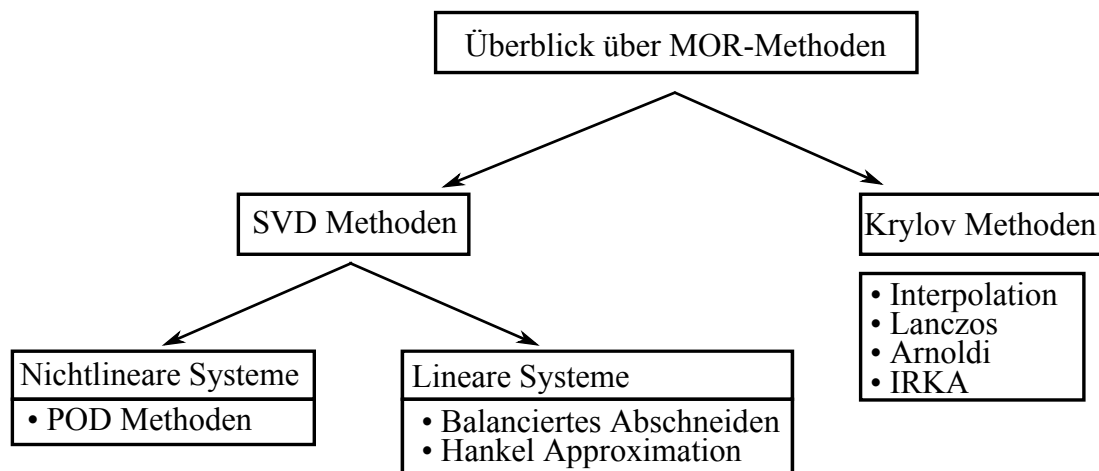


Bild 2-1: Übersicht über die beiden Hauptzweige der MOR-Verfahren (angelehnt an [Ant05])

Ein gegebenes lineares System durch ein System mit weniger Zustandsgleichungen anzunähern, wurde vor fast 50 Jahren vorgeschlagen. In einer der ersten Arbeiten konstruiert DAVISON im Jahr 1966 ein reduziertes System, das die wichtigsten Eigenwerte des Originalsystems beibehält [Dav66]. Diese Idee, heute als modales Abschneiden bezeichnet, wird seitdem kontinuierlich weiterentwickelt. Beispielsweise ist von Litz in [Lit79] ein spezielles Maß für die Dominanz der Eigenwerte vorgestellt worden. In vielen Fällen ist das modale Abschneiden den modernen Verfahren aus verschiedenen Gründen unterlegen und besitzt daher keine große Bedeutung mehr, siehe [BZS⁺09] für eine Gegenüberstellung der Methoden. Es finden sich jedoch weiterhin aktuelle Arbeiten auf diesem Gebiet, beispielsweise aus dem Bereich der Mechanik [TG11, SBS⁺09] sowie für nichtlineare Systeme [BDG07].

Die Modellordnungsreduktion ist vor allem durch die Arbeit von MOORE [Moo81] populär geworden. Dort wird erstmals das balancierte Abschneiden vorgestellt, das neben den in dieser Arbeit verwendeten interpolationsbasierten Verfahren das zweite leistungsfähige und weit verbreitete Verfahren der MOR darstellt, siehe Bild 2-1. Es basiert auf einer Singulärwertzerlegung der Gram'schen Matrizen, mit deren Hilfe der für das Ein-, Ausgangsverhalten wichtigste Systemteil identifiziert werden kann. Die Singulärwertzerlegung bildet die Grundlage für eine Reihe weiterer Verfahren. Eng verwandt mit dem balancierten Abschneiden und ebenfalls auf lineare Systeme zugeschnitten ist die Hankel Approximation, siehe [Glo84]. Für nichtlineare Systeme ist vor allem die Proper Orthogonal Decomposition (POD) weit verbreitet. In der vorliegenden Arbeit werden diese SVD Methoden nicht näher betrachtet. Für eine Einführung sei auf [Ant05] sowie

[ASG01] verwiesen. Die SVD Methoden sind weiterhin Gegenstand zahlreicher und intensiver Forschung, siehe z.B. [AHHS10, HRA11, BS11].

Die interpolationsbasierte MOR bildet den zweiten großen Forschungszweig im Bereich linearer Systeme. Im Gegensatz zum modalen und balancierten Abschneiden steht hier die Übertragungsfunktion, d.h. das Ein-, Ausgangsverhalten des Systems im Frequenzbereich, im Mittelpunkt. Die ersten Ideen in diesem Bereich bestanden darin, eine Übertragungsfunktion mit geringerer Ordnung, d.h. mit kleinerem Zähler- und Nennergrad, aus einer gegebenen Übertragungsfunktion hoher Ordnung zu berechnen. Das reduzierte System stellt dann eine Padé-Approximation des Originalsystems dar. In ersten Arbeiten, z.B. [Kie86, KP88], werden die Koeffizienten des Originalsystems und eine Menge an Stützstellen im Frequenzbereich verwendet, um die Koeffizienten der reduzierten Übertragungsfunktion explizit zu berechnen.

Ein ähnlicher Ansatz wird bei der sogenannten Asymptotic Waveform Evaluation verfolgt [PR90]. Hier wird allerdings zunächst die originale Übertragungsfunktion durch eine endliche Potenzreihe angenähert. Die Koeffizienten dieser Potenzreihe, die sogenannten Momente, werden explizit berechnet und über die Lösung eines linearen Gleichungssystems erhält man das reduzierte Modell auch hier in Form einer Padé-Approximation. All diese expliziten Verfahren führen jedoch zu numerisch sehr schlecht konditionierten Problemen bei steigender Ordnung des reduzierten Modells, siehe [SVR08] für weitere Details. Daher spielen derartige explizite Ansätze in aktuellen Forschungsarbeiten keine Rolle mehr.

Der große Durchbruch im Bereich der interpolationsbasierten MOR entstand durch die Erforschung der Zusammenhänge zwischen Krylov-Unterräumen und den Momenten eines linearen Systems. Auf der Basis des Lanczos-Algorithmus, der bereits Jahrzehnte zuvor entwickelt wurde [Lan50, Lan52], entstand die sogenannte Padé-via-Lanczos-Methode [FF95], mit der eine Padé-Approximation berechnet werden kann, ohne die Momente explizit auszurechnen. Diese Reduktionsvariante wurde kontinuierlich weiterentwickelt und ist insbesondere für symmetrische Matrizen sehr gut geeignet. An dieser Stelle sei auf [Gri97] für eine Übersicht über die historische Entwicklung der Lanczos-Methoden sowie auf [WSW06] als eine beispielhafte Anwendung verwiesen.

Eine weitere Verbesserung, die u.a. für nicht-symmetrische Systeme die Passivität garantieren kann, ergab sich durch die explizite Projektion auf die verwendeten Krylov-Unterräume, die zuerst in den Arbeiten [Gri97] und [OC98] vorgeschlagen wurde. Hier wird der ebenfalls schon lange vorher bekannte Arnoldi-Algorithmus [Arn51] zur Berechnung der Krylov-Unterräume verwendet. Die in dieser Arbeit eingesetzten Modellreduktionsverfahren basieren auf neuesten Entwicklungen aus diesem Forschungszweig. In den letzten 10-15 Jahren sind sehr viele Veröffentlichungen aus diesem Bereich entstanden, sodass eine umfassende Darstellung

unmöglich ist. Stellvertretend seien die Übersichtsartikel [ASG01], [Bai02] und [Fre03] genannt, die jeweils einen Überblick über den damaligen Stand der Technik geben. Die aktuellen und für diese Arbeit besonders relevanten Veröffentlichungen werden nachfolgend in den zugehörigen Abschnitten angegeben.

2.2 Ausgangspunkt, Reduktionsansatz und Bewertungskriterien

Der folgende Abschnitt beschreibt die Grundlagen für die in diesem Kapitel vorgestellten Reduktionsverfahren. Weiterhin werden die Grundidee der Modellordnungsreduktion linearer Systeme sowie das grundlegende mathematische Vorgehen unabhängig von einem konkreten Reduktionsverfahren beschrieben. Auf dieser Basis werden dann eine Reihe von Bewertungskriterien zur Beurteilung der Approximationsgüte eines reduzierten Modells eingeführt.

Die in diesem Kapitel beschriebenen Verfahren sind auf lineare dynamische Systeme der Form

$$\begin{aligned} E\dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t). \end{aligned} \tag{2-1}$$

anwendbar. Hierbei stellt $x \in \mathbb{R}^{n_x}$ den Zustandsvektor, $u \in \mathbb{R}^{n_u}$ die Eingangs- und $y \in \mathbb{R}^{n_y}$ die Ausgangsgrößen des Systems dar. Die Dynamik des Systems wird über die Systemmatrizen $E, A \in \mathbb{R}^{n_x \times n_x}$ und die Eingangsmatrix $B \in \mathbb{R}^{n_x \times n_u}$ bestimmt, die Ausgangsgleichung mit Hilfe der Ausgangsmatrix $C \in \mathbb{R}^{n_y \times n_x}$ und der Durchgriffsmatrix $D \in \mathbb{R}^{n_y \times n_u}$. Im Folgenden wird die Durchgriffsmatrix D vernachlässigt, da sie für die Modellordnungsreduktion nur eine untergeordnete Rolle spielt und zumeist unverändert ins reduzierte System übernommen wird. Weiterhin werden nur gewöhnliche Differentialgleichungen betrachtet, bei denen die Matrix E regulär ist.

Die Übertragungsfunktion $G(s) : \mathbb{C}^{n_u} \rightarrow \mathbb{C}^{n_y}$ von (2-1) im Frequenzbereich lässt sich über die Laplace-Transformation der Zustandsgleichungen

$$(sE - A)X(s) = BU(s), \tag{2-2}$$

$$Y(s) = CX(s), \text{ mit } s \in \mathbb{C}, \tag{2-3}$$

herleiten. Man erhält hieraus als Endergebnis

$$G(s) = C(sE - A)^{-1}B. \tag{2-4}$$

Diese lässt sich in eine Reihendarstellung überführen. Dazu sei zunächst $s_0 \in \mathbb{C}$ ein beliebiger Entwicklungspunkt und $\sigma_s = s - s_0$. Eingesetzt in (2-2) ergibt sich

$$(\sigma_s E - (A - s_0 E))X(s) = BU(s). \tag{2-5}$$

Zur Vereinfachung der Schreibweise sei $\tilde{A}(s_0) := (A - s_0 E)$. Wird nun (2-5) von links mit $\tilde{A}^{-1}(s_0)$ multipliziert und nach $X(s)$ aufgelöst, so erhält man

$$X(s) = - \left(I_{n_x} - \sigma_s \tilde{A}^{-1}(s_0) E \right)^{-1} \tilde{A}^{-1}(s_0) B U(s). \quad (2-6)$$

Mit Hilfe der Neumannschen Reihe [Wer05]

$$\left(I_{n_x} - \sigma_s \tilde{A}^{-1}(s_0) E \right)^{-1} = \sum_{k=0}^{\infty} \left(\sigma_s \tilde{A}^{-1}(s_0) E \right)^k$$

ergibt sich eine Reihendarstellung für die äußere inverse Matrix in (2-6). Setzt man abschließend diese Reihe in (2-6) und (2-3) ein, ist die gesamte Übertragungsfunktion in die Reihendarstellung

$$G(s) = \sum_{k=0}^{\infty} \underbrace{-C \left(\tilde{A}^{-1}(s_0) E \right)^k \tilde{A}^{-1}(s_0) B}_{M_k(s_0)} (s - s_0)^k = \sum_{k=0}^{\infty} M_k(s_0) (s - s_0)^k \quad (2-7)$$

überführt. Die Koeffizienten $M_k(s_0)$ werden als Momente bezeichnet und spielen für die in dieser Arbeit verwendeten Reduktionsverfahren eine entscheidende Rolle. Für Eingrößensysteme handelt es sich bei den Momenten um komplexwertige, skalare Größen. Bei Mehrgrößensystemen sind die Momente rechteckige Matrizen der Dimension $n_y \times n_u$.

Alle in dieser Arbeit verwendeten Verfahren zur Modellreduktion basieren auf einer Projektion des Originalsystems (2-1) mit Hilfe von zwei rechteckigen Projektionsmatrizen $V, W \in \mathbb{R}^{n_x \times q}$ zur Erzeugung des reduzierten Systems des Ordnung q . Dazu ersetzt man den Zustandsvektor x durch die Projektion

$$x = V x_r$$

mit dem reduzierten Zustandsvektor $x_r \in \mathbb{R}^q$. Eingesetzt in (2-1) ergibt sich hieraus

$$\begin{aligned} EV \dot{x}_r(t) &= AV x_r(t) + Bu(t), \\ y_r(t) &= CV x_r(t). \end{aligned}$$

Die Zustandsgleichung wird nachfolgend mittels W^T projiziert, um erneut ein quadratisches, eindeutig lösbares Differentialgleichungssystem zu erhalten. Das reduzierte System ergibt sich somit als

$$\begin{aligned} W^T EV \dot{x}_r(t) &= W^T AV x_r(t) + W^T Bu(t), \\ y_r(t) &= CV x_r(t). \end{aligned} \quad (2-8)$$

Dieser Reduktionsansatz lässt sich auch geometrisch interpretieren. Die Zustandsgleichung wird auf einen q -dimensionalen Unterraum – aufgespannt von den Spalten in V – projiziert. Diese Projektion erfolgt entlang des orthogonalen Komplements des von den Spalten von W aufgespannten Unterraumes (siehe [PMEL10] für weitere Details zu dieser geometrischen Interpretation).

Zur Abkürzung werden die Matrizen

$$\begin{aligned} E_r &:= W^T E V, & A_r &:= W^T A V, \\ B_r &:= W^T B, & C_r &:= C V \end{aligned}$$

definiert, unter der Annahme, dass $W^T E V$ invertierbar ist. Die Übertragungsfunktion des reduzierten Systems ist hiermit analog zu (2-4) gegeben durch

$$G_r(s) = C_r (sE_r - A_r)^{-1} B_r.$$

Auch sie lässt sich um jeden Entwicklungspunkt $s_0 \in \mathbb{C}$ in eine Reihe

$$G_r(s) = \sum_{k=0}^{\infty} -C_r \left(\tilde{A}_r^{-1}(s_0) E_r \right)^k \tilde{A}_r^{-1}(s_0) B_r (s-s_0)^k = \sum_{k=0}^{\infty} M_{r,k}(s_0) (s-s_0)^k \quad (2-9)$$

entwickeln. Die Momente des reduzierten Systems $M_{r,k}$ besitzen die gleiche Dimension wie die Momente des Originalsystems (2-1), weshalb sie sich als Kenngrößen für die Reduktion eignen. Dies wird im weiteren Verlauf dieser Arbeit bei der Herleitung der Reduktionsverfahren näher erläutert.

Nach der Anwendung von Modellreduktionsverfahren stellt sich stets die Frage nach der Beurteilung des Reduktionsergebnisses. Das generelle Ziel besteht darin, den Ausgang y des Originalsystems möglichst exakt durch den Ausgang y_r des reduzierten Systems zu approximieren. Dabei sollen die Abweichungen nicht nur für bestimmte Eingangsfunktionen, sondern allgemein für möglichst viele Eingangssignale gering sein. In dieser Arbeit werden verschiedene Kriterien zur Bewertung des Reduktionsfehlers verwendet und im Folgenden kurz eingeführt.

Zur Klassifizierung linearer Systeme werden die \mathcal{H}_2 und die \mathcal{H}_∞ Norm verwendet, die beide weit verbreitet sind. An dieser Stelle beschränkt sich die Darstellung auf die Angabe der Definitionen. Eine ausführliche Herleitung ist zum Beispiel in [Ant05] zu finden. Eine kürzere Zusammenfassung kann [Kub08] entnommen werden. Dort befindet sich auch eine Erweiterung der \mathcal{H}_2 -Norm auf instabile Systeme. Alle im Folgenden genannten Definitionen und Abschätzungen entstammen [ABG10].

Der Reduktionsfehler $e(t) = y(t) - y_r(t)$ kann mittels des sogenannten Fehlersystems

$$\begin{aligned} \begin{bmatrix} x \\ x_r \end{bmatrix}^\bullet &= \begin{bmatrix} A & 0 \\ 0 & A_r \end{bmatrix} \begin{bmatrix} x \\ x_r \end{bmatrix} + \begin{bmatrix} B \\ B_r \end{bmatrix} u, \\ e = y - y_r &= \begin{bmatrix} C & -C_r \end{bmatrix} \begin{bmatrix} x \\ x_r \end{bmatrix} \end{aligned} \quad (2-10)$$

abgeschätzt werden. Im Folgenden ist unter $\|G - G_r\|$ die Norm des obigen Fehlersystems (2-10) gemeint.

Die \mathcal{H}_∞ -Norm eines stabilen Systems mit der Übertragungsfunktion G aus (2-4) ist definiert als

$$\|G\|_{\mathcal{H}_\infty} = \max_{\omega \in \mathbb{R}} \|G(j\omega)\|_2 = \max_{\omega \in \mathbb{R}} \sigma_{\max}(G(j\omega)),$$

wobei σ_{\max} den größten Singulärwert bezeichnet. Die \mathcal{H}_∞ -Norm beschreibt den durchschnittlichen Fehler von $y(t) - y_r(t)$ im Zeitbereich, da gezeigt werden kann, dass

$$\int_0^\infty \|y(t) - y_r(t)\|_2^2 dt \leq \|G - G_r\|_{\mathcal{H}_\infty}^2$$

für alle Eingangsfunktionen mit $\int_0^\infty \|u(t)\|_2^2 dt \leq 1$ gilt.

Ist man statt der durchschnittlichen Abweichung daran interessiert, dass der Fehler $y(t) - y_r(t)$ zu jedem Zeitpunkt t möglichst klein ist, so ist die \mathcal{H}_2 -Norm geeignet. Sie ist definiert als

$$\|G\|_{\mathcal{H}_2} = \left(\frac{1}{2\pi} \int_{-\infty}^\infty \|G(j\omega)\|_F^2 d\omega \right)^{1/2}.$$

Hierbei bezeichnet $\|\cdot\|_F$ die Frobenius-Norm, die ihrerseits definiert ist als

$$\|G(s)\|_F^2 = \sum_{k=1}^{\min(n_y, n_u)} \sigma_k^2(G(s)) = \text{spur}(G^*(s)G(s)) = \sum_{i,j} |G_{ij}(s)|^2,$$

wobei mit G^* die adjungierte Matrix und mit σ_k erneut die Singulärwerte der Übertragungsmatrix G bezeichnet sind. Analog zur \mathcal{H}_∞ -Norm lässt sich eine Fehlerabschätzung angeben:

$$\max_{t \geq 0} \|y(t) - y_r(t)\|_\infty \leq \|G - G_r\|_{\mathcal{H}_2}.$$

Neben dem absoluten Fehler ist oftmals der relative Fehler

$$E_{\text{rel}} = \frac{\|G - G_r\|}{\|G\|}$$

aussagekräftiger, da er die Abweichungen aus dem Fehlersystem auf die Norm des Originalsystems bezieht. Insbesondere für algorithmische Umsetzungen, die unabhängig von konkreten Systemen sein sollen, ist stets eine relative Betrachtung erforderlich.

Diese beiden auf dem Frequenzbereich basierenden Normen zusammen mit den angeführten Abschätzungen stellen leistungsfähige und effizient umsetzbare Kriterien dar. In manchen Fällen wird es aber darüber hinaus notwendig sein, die Approximationsgüte des reduzierten Systems auch explizit im Zeitbereich zu untersuchen. Hierzu sind einerseits Sprung- und Impulsanregungen geeignet, da sie die gesamte Systemdynamik anregen und beispielsweise den stationären Fehler aufzeigen. Auf diese Weise können – oft über eine grafische Auswertung – Schwächen des reduzierten Systems erkannt werden, die beispielsweise aus unterschiedlich skalierten Ein- und Ausgängen resultieren. Andererseits werden gelegentlich auch komplexere Simulationen zur Bewertung des reduzierten Modells genutzt. Beispielsweise wurde in [KWTD11] das reduzierte System in einem hierarchischen Modell eingebunden und eine stochastische Anregung des resultierenden Gesamtsystems simuliert und bewertet.

2.3 Manuelle rationale Interpolation

Bei der manuellen rationalen Interpolation wird das reduzierte System, wie im vorhergehenden Abschnitt beschrieben, über eine Projektion gebildet. Die rationale Interpolation stellt diesbezüglich eine Möglichkeit zur Berechnung der Projektionsmatrizen dar. Als Kenngrößen für die Reduktion werden die Momente aus (2-7) bzw. (2-9) verwendet. Die Idee besteht darin, möglichst viele der Momente des Originalsystems durch das reduzierte System abzugleichen. Dazu wird im Rahmen dieses Abschnitts der bereits in der Einführung erwähnte Arnoldi-Algorithmus vorgestellt. Obwohl dieser in seiner Grundform, siehe [Arn51], bereits seit mehr als 60 Jahren bekannt ist, wird er für die Modellordnungsreduktion erst seit etwa 20 Jahren eingesetzt.

Im Bereich der Regelungstechnik wurde der Arnoldi-Algorithmus zunächst, etwa seit den 1980er Jahren, zur Untersuchung der Steuerbarkeit und Beobachtbarkeit verwendet, z.B. in [BG84] zur Berechnung des steuerbaren Unterraumes. Bis in die 1990er Jahre blieb dies der Fall. In Arbeiten zur MOR wird er zwar gelegentlich erwähnt, beispielsweise in [Doo92], aber der Fokus liegt hier auf dem ähnlichen und bereits in der Kapiteleinführung erwähnten Lanczos-Algorithmus.

Eine erste Anwendung zur Modellordnungsreduktion findet sich in [SKW96], allerdings wird in dieser Arbeit noch keine explizite Projektion zur Berechnung des reduzierten Systems durchgeführt. Diese explizite Projektion auf geeignete Krylov-Unterräume, um einen Abgleich der Momente zu erzielen, wird in [Gri97]

vorgestellt. Hier wird zudem bereits die Verwendung mehrerer Entwicklungspunkte beschrieben, die im Folgenden noch näher erläutert wird. Diese Dissertation stellt eine wichtige Grundlage für die in dieser Arbeit behandelten theoretischen Zusammenhänge dar. Seit dieser Zeit hat sich der Arnoldi-Algorithmus zu einem weit verbreiteten und etablierten Hilfsmittel für die Modellordnungsreduktion entwickelt.

Die Variante des in dieser Arbeit weiterentwickelten Arnoldi-Algorithmus für parametrische Systeme, ausführlich beschrieben in Kapitel 3, basiert wesentlich auf dem Übersichtsartikel [LS04]. Weiterhin sind vor allem die Abbruch- und Auswahlkriterien aus [SLBK05] in die vorliegende Arbeit eingeflossen. Zusammen mit den beiden Dissertationen [Sal05] und [Eid09] bilden sie das Grundgerüst, das zur parametrischen MOR erweitert wird. Die wesentlichen theoretischen Grundlagen und einige Details zum Arnoldi-Algorithmus werden innerhalb dieses Abschnitts hergeleitet.

2.3.1 Impliziter Momentenabgleich

Den Kern des Momentenabgleichs, d.h. der Berechnung eines reduzierten Systems, dessen Momente denen des Originalsystems entsprechen, bilden die Krylov-Unterräume. Allgemein ist ein Krylov-Unterraum definiert als

$$\mathcal{K}(A_{\mathcal{K}}, b_{\mathcal{K}}) = \text{span} \{b_{\mathcal{K}}, A_{\mathcal{K}}b_{\mathcal{K}}, A_{\mathcal{K}}^2b_{\mathcal{K}}, \dots\}$$

mit einer konstanten Matrix $A_{\mathcal{K}} \in \mathbb{R}^{n_1 \times n_1}$ und einem konstanten Vektor $b_{\mathcal{K}} \in \mathbb{R}^{n_1}$. Anstelle dieses Vektors, der auch Startvektor genannt wird, kann auch eine Matrix $B_{\mathcal{K}} \in \mathbb{R}^{n_1 \times n_2}$ verwendet werden. Dann wird

$$\mathcal{K}(A_{\mathcal{K}}, B_{\mathcal{K}}) = \text{span} \{B_{\mathcal{K}}, A_{\mathcal{K}}B_{\mathcal{K}}, A_{\mathcal{K}}^2B_{\mathcal{K}}, \dots\}$$

als Block-Krylov-Unterraum bezeichnet. Für die MOR wird zusätzlich eine positive Zahl $q \in \mathbb{N}$ genutzt und die Sequenz $A_{\mathcal{K}}^i B_{\mathcal{K}}$ bei $q - 1$ abgebrochen. Hiermit erhält man den q -ten Krylov-Unterraum

$$\mathcal{K}_q(A_{\mathcal{K}}, B_{\mathcal{K}}) = \text{span} \{B_{\mathcal{K}}, A_{\mathcal{K}}B_{\mathcal{K}}, A_{\mathcal{K}}^2B_{\mathcal{K}}, \dots, A_{\mathcal{K}}^{q-1}B_{\mathcal{K}}\}. \quad (2-11)$$

Eine Verknüpfung der Krylov-Unterräume mit den Momenten einer Übertragungsfunktion ergibt sich durch die folgenden Aussagen.

Stellen die Spalten der für die Reduktion verwendeten Projektionsmatrix V eine Basis des sogenannten eingangsseitigen Krylov-Unterraumes

$$\mathcal{K}_q(\tilde{A}^{-1}(s_0)E, \tilde{A}^{-1}(s_0)B)$$

dar, so stimmen die ersten q Momente zwischen Originalsystem und reduziertem System am Entwicklungspunkt s_0 überein, d.h.

$$M_j(s_0) = M_{r,j}(s_0), \quad 0 \leq j \leq q-1,$$

in der Notation des vorherigen Abschnitts. Die andere Projektionsmatrix W kann für die Reduktion beliebig gewählt werden mit der Einschränkung, dass $W^T E V$ regulär sein muss. Im Fall eines Eingrößensystems besitzt dann auch das reduzierte System die Ordnung q . Andernfalls entspricht die Ordnung der Anzahl der Basisvektoren, die zur Aufspannung des eingangsseitigen Krylov-Unterraumes benötigt werden.

Bilden umgekehrt die Spalten der Projektionsmatrix W eine Basis des ausgangsseitigen Krylov-Unterraumes

$$\mathcal{K}_q(\tilde{A}^{-T}(s_0)E^T, \tilde{A}^{-T}(s_0)C),$$

so stimmen ebenfalls die ersten q Momente überein. Analog zum ersten Fall kann die zweite Projektionsmatrix V beliebig sein, solange die Regularität von $W^T E V$ sichergestellt ist.

Sind beide Bedingungen erfüllt, d.h. V ist Basis von $\mathcal{K}_{q_1}(\tilde{A}^{-1}(s_0)E, \tilde{A}^{-1}(s_0)B)$ und W ist Basis von $\mathcal{K}_{q_2}(\tilde{A}^{-T}(s_0)E^T, \tilde{A}^{-T}(s_0)C)$, so addieren sich die übereinstimmenden Momente. Es werden dann insgesamt $q_1 + q_2$ Momente abgeglichen, d.h. $M_j = M_{r,j}$ für $0 \leq j \leq q_1 + q_2 - 1$. In diesem Fall spricht man von einem zweiseitigen Verfahren. Die beiden vorhergehenden Fälle werden entsprechend als einseitige Verfahren bezeichnet. Formale Beweise zur Gültigkeit des Momentenabgleichs sind beispielsweise in [Sal05] zu finden.

Wird nur ein Entwicklungspunkt s_0 betrachtet, so entsteht eine Art Padé-Approximation des Originalsystems, da die Übertragungsfunktion und ihre Ableitungen an dem festgelegten Punkt s_0 mit dem reduzierten System übereinstimmen.

Es ist ohne weiteres möglich, mehrere Entwicklungspunkte zu verwenden. Bilden die Matrizen V_1, \dots, V_k jeweils eine Basis der eingangsseitigen Krylov-Unterräume zu den Entwicklungspunkten s_1, \dots, s_k und W_1, \dots, W_k entsprechend eine Basis der ausgangsseitigen Krylov-Unterräume, so stimmt die jeweilige Anzahl der Momente an jedem Entwicklungspunkt überein. Die einzige Voraussetzung besteht darin, dass die Gesamtprojektionsmatrizen V und W alle Basisvektoren beinhalten, d.h.

$$V = \text{span} \{V_1, \dots, V_k\}, \quad W = \text{span} \{W_1, \dots, W_k\}.$$

Die Verwendung von Entwicklungspunkten $s_0 \neq 0$ sowie von mehreren Entwicklungspunkten bei der MOR wird als rationale Interpolation bezeichnet und stellt

die allgemeinste Form der MOR auf Basis eines impliziten Momentenabgleichs dar.

Die wesentliche Stärke der hier vorgestellten Reduktion ist darin begründet, dass die Momente mit jeder beliebigen Basis der entsprechenden Krylov-Unterräume abgeglichen werden. Insbesondere ist eine explizite Berechnung der Momente zur Bestimmung von V und W nicht erforderlich. Dies ist bedeutsam, da die Berechnung der Momente ein numerisch schlecht-konditioniertes Problem darstellt und frühere auf den Momenten basierende Methoden an dieser Problematik scheiterten.

Die Ordnung des reduzierten Modells folgt direkt aus der Anzahl der abgeglichenen Momente bzw. der Anzahl der verwendeten Basisvektoren. Bei Mehrgrößensystemen werden mehrere Basisvektoren benötigt, um ein Moment abzugleichen. Für den eingangsseitigen Krylov-Unterraum entspricht diese Anzahl der Zahl der Eingänge, für den ausgangsseitigen Krylov-Unterraum ist die Anzahl der Ausgänge maßgebend. Für die praktische Umsetzung spielt dies jedoch eine untergeordnete Rolle. Es ist ohne Probleme möglich, einzelne Basisvektoren zu ergänzen, so dass beliebige Ordnungen des reduzierten Modells entstehen. Dies entspricht dem Abgleich einzelner Spalten (für den eingangsseitigen Krylov-Unterraum) bzw. einzelner Zeilen (für den ausgangsseitigen Krylov-Unterraum) der Momente. Eine Alternative hierzu, die sogenannte tangentielle Interpolation, wird im nächsten Abschnitt 2.4 näher betrachtet.

2.3.2 Arnoldi-Algorithmus

Die Basisvektoren werden im Rahmen dieser Arbeit mit dem Arnoldi-Algorithmus berechnet. Der grundlegende Ablauf für den Fall eines Krylov-Unterraums (2-11) mit einem einzelnen Vektor b_K als Startvektor ist in Algorithmus 1 dargestellt.

Aus dem Algorithmus wird bereits deutlich, dass die Vorgabe der gewünschten Ordnung des reduzierten Modells einen wesentlichen Steuerungsparameter darstellt. Besondere Aufmerksamkeit erfordert aber auch die Bedingung $\|v_i\| > 0$ in Schritt 4. Diese gibt an, ob der neue Basisvektor linear unabhängig zu den bisherigen Basisvektoren ist. Bei der numerischen Umsetzung wird an dieser Stelle eine Schranke ε_D verwendet. Ist das Residuum $\|v_i\|$ der Orthogonalisierung, d.h. der Anteil neuer Informationen für die Basis, kleiner als die Schranke, wird der Kandidat verworfen und die Iteration beendet. Somit ergibt sich über die Wahl von ε_D eine weitere Steuerungsmöglichkeit für den Algorithmus, mit der die maximale Ordnung des reduzierten Systems beeinflusst werden kann. Die durch den Algorithmus berechneten Basisvektoren bilden eine Orthonormalbasis des Krylov-Unterraums. Dies trägt wesentlich zum gutmütigen numerischen Verhalten der MOR auf Basis des Arnoldi-Algorithmus bei.

Algorithmus 1 Arnoldi-Algorithmus (Grundform mit individuellem Startvektor, [LS04])

Input: Startvektor b_K , quadratische Matrix A_K und Ordnung q

```

1: Initialisierung:
2:  $v_1 = \frac{b_K}{\|b_K\|}$ 
3:  $V = v_1$ 
4:  $i = 2$ 
5: while  $i \leq q$  do
6:    $v_i = A_K v_{i-1}$  (nächster Basisvektor)
7:    $v_i = v_i - VV^T v_i$  (Orthogonalisierung)
8:   if  $\|v_i\| > 0$  then
9:      $v_i = \frac{v_i}{\|v_i\|}$  (Normalisierung)
10:     $V = [V, v_i]$ 
11:   else
12:     STOP
13:   end if
14:    $i = i + 1$ 
15: end while

```

Output: Orthonormalbasis V des Krylov-Unterraums (2-11)

Die drei Grundbestandteile des Arnoldi-Algorithmus *rekursive Berechnung des nächsten Basisvektors mittels einer Matrix-Vektor-Multiplikation, Orthogonalisierung* sowie *Normalisierung* bleiben auch bei der Reduktion von Mehrgrößensystemen erhalten. Anstelle des Startvektors b_K steht nun eine Matrix B_K zur Verfügung, deren Spalten einzeln verwendet werden. Durch die Reihenfolge der Verwendung ergibt sich eine weitere Freiheit des Algorithmus. Im Rahmen dieser Arbeit werden die Startvektoren in der Reihenfolge ihrer Anordnung in B_K verwendet. Eine aufwändigere Selektion über ein zusätzliches Dominanzmaß ist in [SLBK05] zu finden. Die Abbruchschranke ε_D wird auch im Mehrgrößensystem verwendet. Ist ein Residuum kleiner als ε_D , wird der jeweilige Kandidat aus der Iteration entfernt. Der Algorithmus wird mit den übrigen Basisvektoren fortgesetzt. Dieses Verwerfen einzelner Vektoren während der Iteration wird auch als Deflation bezeichnet. Die Deflation trägt zur numerischen Robustheit sowie zu einer höheren Approximationsgüte des reduzierten Modells bei.

Mit dem Arnoldi-Algorithmus lassen sich beide Projektionsmatrizen V und W gleich effizient berechnen. Für eine zweiseitige MOR braucht der Algorithmus daher lediglich zweimal, für V und für W , nacheinander ausgeführt werden. Als einzige Bedingung müssen die Dimensionen, d.h. die Anzahl der Basisvektoren, übereinstimmen. Dies stellt jedoch keine große Einschränkung bei der praktischen Realisierung dar. Sollte aufgrund der Deflation eine der beiden Projektionsma-

trizen weniger Spalten enthalten als die andere, so kann von der größeren Matrix die entsprechende Differenzanzahl vernachlässigt werden.

Bei den in dieser Arbeit untersuchten Anwendungsfällen hat die Abbruchschranke ε_D , für deren Wahl es keinerlei Kriterien gibt, nur eine untergeordnete Rolle gespielt. Für die Approximationsgüte des reduzierten Modells war die Vorgabe der Ordnung fast immer erforderlich und hatte generell eine größere Auswirkung auf das Reduktionsergebnis.

Bei der rationalen Interpolation werden wie bereits dargestellt mehrere Entwicklungspunkte verwendet. Auch dies lässt sich durch eine mehrfache Anwendung des Arnoldi-Algorithmus realisieren. Zu jedem Entwicklungspunkt s_i wird zunächst die zugehörige Basis V_i bzw. W_i berechnet. Anschließend werden die Gesamtprojektionsmatrizen mit Hilfe einer Orthonormalisierung,

$$V = \text{orth}(V_1, \dots, V_k), \quad W = \text{orth}(W_1, \dots, W_k),$$

bestimmt. Die Dimension des reduzierten Systems ergibt sich als Summe der Spalten der V_i bzw. W_i unter der Annahme, dass diese linear unabhängig sind.

Auch wenn sich bei der numerischen Umsetzung der rationalen Interpolation wenig prinzipielle Schwierigkeiten ergeben, so ist vor allem die hohe Zahl der die Reduktion beeinflussenden Parameter von großem Nachteil. Es existieren keine allgemeingültigen Heuristiken oder Verfahren, um die Anzahl sowie die optimalen Werte der Entwicklungspunkte festzulegen. Die Tatsache, dass auch nicht reelle Entwicklungspunkte in Form konjugiert komplexer Paare genutzt werden können, erschwert die Festlegung zusätzlich. Neben der Lage und der Anzahl der Entwicklungspunkte ist zusätzlich noch die Anzahl der Basisvektoren nahezu frei vorgebar, wodurch sich selbst bei einer festgesetzten Ordnung des reduzierten Modells eine unüberschaubare Menge von Kombinationsmöglichkeiten ergibt.

Eine Methode zur Festlegung eines optimalen Entwicklungspunktes wurde in [Eid09] auf der Basis von Laguerre-Funktionen entwickelt. Für Eingrößensysteme konnte eine explizite Berechnungsvorschrift angegeben werden. Eine Verallgemeinerung auf Mehrgrößensysteme existiert jedoch nicht, weshalb die optimalen Entwicklungspunkte nach EID insgesamt für diese Arbeit keine Rolle spielen.

Trotz der Problematik, die aus den vielen Freiheitsgraden resultiert, konnten mit der manuellen rationalen Interpolation gute Ergebnisse erzielt werden. Eine wesentliche Verbesserung, insbesondere in der Automatisierbarkeit stellt jedoch die im nächsten Abschnitt beschriebene H_2 -optimale Interpolation dar.

2.4 H_2 -optimale tangentielle Interpolation

In diesem Kapitel wird mit der \mathcal{H}_2 -optimalen Interpolation eine der neuesten Methoden aus dem Bereich der interpolationsbasierten MOR vorgestellt. Der wesentliche Vorteil gegenüber der manuellen rationalen Interpolation besteht zum

einen darin, dass notwendige Bedingungen für ein optimales reduziertes System angegeben werden können. Zum anderen ist in den letzten Jahren ein robuster iterativer Algorithmus, der *Iterative Rational Krylov Algorithm (IRKA)*, entwickelt worden. Dieser dient zur iterativen Berechnung der Projektionsmatrizen und führt oftmals zu reduzierten Systemen mit sehr hoher Approximationsgüte. Beides, die Optimalitätsbedingungen und der IRKA sind in [GAB08] zu finden. In [BG09] sind Verallgemeinerungen zur Erhaltung struktureller Eigenschaften im reduzierten System beschrieben. Beispielsweise wird die Reduktion von Systemen zweiter Ordnung behandelt. Derartige Aspekte spielen für die vorliegende Arbeit keine Rolle. Eine Einführung in den aktuellen Stand der Forschung zur \mathcal{H}_2 -optimalen Interpolation gibt [ABG10]. Dieses Buchkapitel dient auch als Grundlage für die Ausführungen in den folgenden Abschnitten.

Die \mathcal{H}_2 -optimale tangentielle Interpolation basiert auf der tangentialen Interpolation (siehe [GVD04]), die aus diesem Grund zunächst in Abschnitt 2.4.1 erläutert wird. Anschließend widmet sich Abschnitt 2.4.2 den theoretischen Grundlagen der \mathcal{H}_2 -optimalen Interpolation und Abschnitt 2.4.3 beschreibt abschließend den IRKA. In neuesten Arbeiten finden sich Alternativen zum IRKA. Beispielsweise beschreibt [XZ11] einen ähnlichen Algorithmus, der jedoch auf der Lösung bestimmter Sylvestergleichungen basiert und daher aufwändiger ist. Einen Algorithmus, der die Entwicklungspunkte bei der rationalen Interpolation adaptiv wählt, wird in [DS11] vorgestellt. Allerdings handelt es sich hierbei um die Berechnung einseitiger Projektionen, die zudem nur auf Eingrößensysteme angewendet werden kann. Daher ist auch diese Variante nicht geeignet für die vorliegende Arbeit.

2.4.1 Tangentielle Interpolation

Die tangentielle Interpolation basiert wie die in Abschnitt 2.3 beschriebene manuelle rationale Interpolation auf der Interpolation der Übertragungsfunktion des Originalsystems. Die Zielsetzungen beider Varianten unterscheiden sich jedoch. Bei der manuellen rationalen Interpolation werden zu Beginn ein bzw. einige Entwicklungspunkte für den Momentenabgleich ausgewählt. Dann wird die Anzahl der abzugleichenden Momente erhöht bis eine zufriedenstellende Approximation durch das reduzierte System erzielt wird. Bei der tangentialen Interpolation sucht man eine Sequenz von Entwicklungspunkten, an denen üblicherweise nur die nullten Momente, d.h. die Werte der Übertragungsfunktionen selbst, abgeglichen werden. Die Ordnung des reduzierten Modells korrespondiert daher mit der Anzahl der Entwicklungspunkte innerhalb der Sequenz.

Bei diesem Ansatz besteht zunächst erneut das Problem, dass im Falle eines Mehrgrößensystems die Momente Matrizen darstellen und somit mehrere Projektionsvektoren erforderlich sind, um einen Abgleich zu gewährleisten. Die Lösungsidee

bei der tangentialen Interpolation besteht nun darin, statt der gesamten Momente lediglich eine bestimmte Richtung anzugeben, in der die Momente übereinstimmen. Mathematisch ausgedrückt bedeutet dies, dass Paare $[\sigma_1, \dots, \sigma_q]$, $[\tilde{b}_1, \dots, \tilde{b}_q]$ bzw. $[\mu_1, \dots, \mu_q]$, $[\tilde{c}_1, \dots, \tilde{c}_q]$ aus jeweils einer Sequenz von Entwicklungspunkten $\sigma_i, \mu_i \in \mathbb{C}$ mit zugehörigen Tangentialvektoren $\tilde{b}_i \in \mathbb{C}^{n_u}$, $\tilde{c}_i \in \mathbb{C}^{n_y}$ gebildet werden. Für die Übertragungsfunktionen wird dann gefordert, dass

$$G(\sigma_i)\tilde{b}_i = G_r(\sigma_i)\tilde{b}_i \quad \text{und} \quad \tilde{c}_i^T G(\mu_i) = \tilde{c}_i^T G_r(\mu_i), \quad 1 \leq i \leq q$$

gilt. Da $G(\mu_i) = M_0(\mu_i)$ und $G_r(\mu_i) = M_{r,0}(\mu_i)$ gilt, lässt sich diese Bedingung auch über die Momente ausdrücken:

$$M_0(\sigma_i)\tilde{b}_i = M_{r,0}(\sigma_i)\tilde{b}_i \quad \text{und} \quad \tilde{c}_i^T M_0(\mu_i) = \tilde{c}_i^T M_{r,0}(\mu_i), \quad 1 \leq i \leq q.$$

Die tangentiale Interpolation kann ebenfalls mit Hilfe geeigneter Projektionsmatrizen erreicht werden. Das Hauptresultat besteht aus den folgenden drei Aussagen.

- (I) Falls $(\sigma E - A)^{-1}B\tilde{b} \in \text{span}(V)$, dann gilt $M_0(\sigma)\tilde{b} = M_{r,0}(\sigma)\tilde{b}$.
- (II) Falls $(\tilde{c}^T C(\mu E - A)^{-1})^T \in \text{span}(W)$, dann gilt $\tilde{c}^T M_0(\mu) = \tilde{c}^T M_{r,0}(\mu)$.
- (III) Falls (I) und (II) gelten und $\sigma = \mu$ gewählt ist, so gilt zusätzlich für die Ableitungen der Übertragungsfunktionen

$$\tilde{c}^T G'(\sigma)\tilde{b} = \tilde{c}^T M_1(\sigma)\tilde{b} = \tilde{c}^T M_{r,1}(\sigma)\tilde{b} = \tilde{c}^T G'_r(\sigma)\tilde{b}.$$

Ein Beweis ist beispielsweise in [ABG10] nachzulesen. Erneut bezeichnen $\sigma, \mu \in \mathbb{C}$ zwei Entwicklungs- bzw. Interpolationspunkte und $\tilde{b} \in \mathbb{C}^{n_u}$ sowie $\tilde{c} \in \mathbb{C}^{n_y}$ die zugehörigen Tangentialvektoren. Mit V und W sind analog zum vorhergehenden Unterkapitel die Projektionsmatrizen gemeint.

Die letzte Aussage überträgt das Resultat, dass sich die abgeglichenen Momente bei zweiseitigen Verfahren addieren, auf die tangentiale Interpolation. Aus (I) und (II) ergeben sich auch direkt die Berechnungsvorschriften für die Projektionsmatrizen. Sind die Interpolationspunkte $\{\sigma_i\}_{i=1}^q$ und $\{\mu_i\}_{i=1}^q$ sowie die zugehörigen Tangentialvektoren $\{\tilde{b}_i\}_{i=1}^q$ und $\{\tilde{c}_i\}_{i=1}^q$ gegeben, so ergeben sich V und W als

$$V = [(\sigma_1 E - A)^{-1}B\tilde{b}_1, \quad \dots, \quad (\sigma_q E - A)^{-1}B\tilde{b}_q], \quad W = \begin{bmatrix} \tilde{c}_1^T C(\mu_1 E - A)^{-1} \\ \vdots \\ \tilde{c}_q^T C(\mu_q E - A)^{-1} \end{bmatrix}.$$

Auch hier ist, wie bei der manuellen rationalen Interpolation, keine explizite Berechnung der abzugleichenden Größen erforderlich, die ebenfalls schlecht konditioniert wäre. Weiterhin spielt die Anzahl der Ein- und Ausgänge nur noch eine

untergeordnete Rolle für die Dimension des reduzierten Systems. Dies ist ein wesentlicher Unterschied zu der manuellen rationalen Interpolation, bei der mit steigender Zahl der Ein- bzw. Ausgänge, die Ordnung des reduzierten Modells ebenfalls anstieg, um beispielsweise die Momente erster Ordnung abzugleichen. Ungelöst ist an dieser Stelle allerdings noch, wie die Interpolationspunkte und die Tangentialvektoren zu wählen sind. Generell sind auch komplexe Vektoren zulässig. Man ist zwar an reellen reduzierten Systemen interessiert, diese lassen sich aber erzeugen, falls die Interpolationspunkte und Tangentialvektoren bzgl. der komplexen Konjugation abgeschlossen¹ sind, siehe z.B. [Gri97] für weitere Details.

2.4.2 Theoretische Grundlagen der \mathcal{H}_2 -optimalen Interpolation

Das Ziel der \mathcal{H}_2 -optimalen Interpolation besteht darin, reduzierte Systeme zu berechnen, die optimal im Sinne der in Abschnitt 2.2 definierten \mathcal{H}_2 -Norm sind. Gesucht ist also ein reduziertes System, sodass für eine festgelegte Ordnung q

$$\|G - G_r\|_{\mathcal{H}_2} = \min_{\tilde{G}_r} \left(\|G - \tilde{G}_r\|_{\mathcal{H}_2} \mid \dim(\tilde{G}_r) = q, \tilde{G}_r \text{ ist stabil} \right) \quad (2-12)$$

erfüllt ist. Es handelt sich hierbei um ein nichtkonvexes Optimierungsproblem, dessen Lösung im Allgemeinen nicht geschlossen angegeben werden kann. Es kann aber eine notwendige Bedingung für die Lösung dieses Optimierungsproblems hergeleitet werden, was wesentlich zur Leistungsfähigkeit der \mathcal{H}_2 -optimalen Interpolation beiträgt.

Im Folgenden wird eine weitere Reihenentwicklung der Übertragungsfunktion basierend auf der Modalform benötigt. Ist ein lineares System (2-1) mit einfachen Eigenwerten $\lambda_1, \dots, \lambda_{n_x}$ gegeben, so lassen sich die Systemmatrizen E und A diagonalisieren. Dazu löst man das verallgemeinerte Eigenwertproblem

$$Ax = \lambda Ex$$

und bestimmt hieraus die Eigenwerte sowie die Linkseigenvektoren Υ^* mit

$$\Upsilon^* A = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_{n_x} \end{bmatrix} \Upsilon^* E = \Lambda \Upsilon^* E$$

und die Rechtseigenvektoren Ψ mit

$$A\Psi = E\Psi\Lambda.$$

¹Eine Menge M von Vektoren ist abgeschlossen bzgl. der komplexen Konjugation, wenn für jeden Vektor $m \in M$ auch der konjugiert komplexe Vektor $\bar{m} \in M$ liegt.

Mit Hilfe der Eigenvektoren lässt sich unter Berücksichtigung einer entsprechenden Normierung, das System nun auf Modal- bzw. Diagonalform transformieren, denn es gilt

$$\Upsilon^* A \Psi = \Lambda \text{ und } \Upsilon^* E \Psi = I_{n_x}.$$

Das System besitzt dann die Zustandsdarstellung

$$\begin{aligned} \dot{z} &= \Lambda z + \hat{B}u, \\ y &= \hat{C}z \end{aligned}$$

und die Übertragungsfunktion

$$G(s) = \sum_{i=1}^{n_x} \frac{\hat{c}_i \hat{b}_i^T}{s - \lambda_i}.$$

Dabei bezeichnen \hat{c}_i die Spalten von $\hat{C} = C\Psi$ und \hat{b}_i^T die Zeilen von $\hat{B} = \Upsilon^* B$.

Die notwendige Bedingung für eine \mathcal{H}_2 -optimale Interpolation lautet nun folgendermaßen: G sei die Übertragungsfunktion eines stabilen Systems und G_r sei eine Übertragungsfunktion, die (2-12) erfüllt. Weiterhin seien mit $\lambda_1, \dots, \lambda_q$ die Pole des reduzierten Systems bezeichnet und G_r sei durch die Summe $G_r(s) = \sum_{k=1}^q \frac{\hat{c}_k \hat{b}_k^T}{s - \lambda_k}$ gegeben. Dann sind die folgenden drei Gleichungen erfüllt:

- (I) $M_0(-\lambda_k) \hat{b}_k = M_{r,0}(-\lambda_k) \hat{b}_k$
- (II) $\hat{c}_k^T M_0(-\lambda_k) = \hat{c}_k^T M_{r,0}(-\lambda_k)$
- (III) $\hat{c}_k^T M_1(-\lambda_k) \hat{b}_k = \hat{c}_k^T M_{r,1}(-\lambda_k) \hat{b}_k$

Das bedeutet, die notwendigen Bedingungen für ein \mathcal{H}_2 -optimales reduziertes System führen auf eine spezielle tangentielle Interpolation. Als Entwicklungs- bzw. Interpolationspunkte sind die an der imaginären Achse gespiegelten Eigenwerte des reduzierten Systems zu wählen und als Tangentialvektoren die Eingangs- bzw. Ausgangsvektoren \hat{c}_k bzw. \hat{b}_k aus der Modalform des reduzierten Systems. Der Beweis der notwendigen Bedingungen folgt aus funktionalanalytischen Eigenschaften der \mathcal{H}_2 -Norm bzw. des zugehörigen Hilbertraumes und diversen Resultaten aus der Funktionentheorie, insbesondere einer Anwendung des Residuensatzes. Eine detaillierte Darstellung ist in [ABG10] zu finden.

Die notwendigen Bedingungen sind für die praktische Berechnung von Projektionsmatrizen zunächst wenig nützlich, da sowohl die Eigenwerte als auch die Tangentialvektoren erst durch das reduzierte System selbst, also nach der Berechnung der Projektionsmatrizen bekannt sind. Optimale reduzierte Systeme bzw. Kandidaten für optimale reduzierte Systeme lassen sich demnach nicht direkt aus

den notwendigen Bedingungen berechnen. Diese Problematik wird im nächsten Abschnitt noch genauer behandelt.

Zuvor sei zu den notwendigen Bedingungen noch angemerkt, dass diese im Allgemeinen nicht hinreichend sind. Da das eingangs formulierte Ziel (2-12) zu einem nichtkonvexen Optimierungsproblem führt, ist es ohne weiteres möglich, dass mehrere stabile reduzierte Systeme einer festen Ordnung q die notwendigen Bedingungen erfüllen, ohne das globale Minimum gemäß (2-12) zu sein. Es handelt sich dann um lokale Minima, die die \mathcal{H}_2 -Norm lediglich innerhalb einer gewissen Umgebung minimieren.

2.4.3 Iterative Berechnung des reduzierten Modells

Die notwendigen Bedingungen (2-12) für ein \mathcal{H}_2 -optimales reduziertes System lassen sich wie im letzten Abschnitt dargestellt mittels tangentialer Interpolation erreichen. Da die Interpolationspunkte die gespiegelten Eigenwerte und die Tangentialvektoren die Residuen der Übertragungsfunktion des reduzierten Systems sind, können sie nicht direkt berechnet werden. Die Grundidee des in diesem Abschnitt vorgestellten *Iterative Rational Krylov Algorithm (IRKA)* besteht darin, die Interpolationspunkte und Tangentialvektoren iterativ anzupassen, bis die Optimalitätsbedingungen erfüllt sind.

Algorithmus 2 beschreibt den Ablauf des IRKA. Ausgehend von initialen Interpolationspunkten und Tangentialvektoren wird ein temporäres reduziertes System berechnet. Dessen Eigenwerte und Eigenvektoren werden zur Konstruktion der neuen Interpolationspunkte und Tangentialvektoren genutzt. Dies wird solange wiederholt, bis sich die Interpolationspunkte im Vergleich zur vorhergehenden Iteration nicht mehr verändern, d.h. der Algorithmus ist konvergiert. Hierfür wird erneut eine Abbruchschranke ε_I verwendet (siehe Zeile 6).

Es ist für den IRKA von essentieller Bedeutung, dass sowohl die Interpolationspunkte, als auch die Tangentialvektoren komplexe Werte annehmen können. Dies führt dazu, dass die temporären reduzierten Systeme (Zeile 7) ebenfalls komplexwertige Systemmatrizen besitzen. Da diese jedoch nur für die Berechnung der Eigenwerte und Eigenvektoren genutzt werden, bringt dies keine Probleme mit sich. Solange die initialen Interpolationspunkte abgeschlossen bzgl. der komplexen Konjugation sind, bleibt dies auch während der Iteration erhalten. Es lässt sich dann am Ende stets ein reelles reduziertes System berechnen, indem die Real- und Imaginärteile der Spalten von V und W als Projektionsvektoren verwendet werden. Dies lässt sich beispielsweise unter Verwendung einer Singulärwertzerlegung implementieren.

Der IRKA hängt, ähnlich wie der Arnoldi-Algorithmus, von einer recht hohen Zahl von Parametern ab. Neben der Ordnung q des reduzierten Modells, die

Algorithmus 2 IRKA – Iterative Rational Krylov Algorithm [ABG10]

Input: Interpolationspunkte $\sigma = [\sigma_1, \dots, \sigma_q]$ und Tangentialvektoren $\hat{b}_i \in \mathbb{C}^{n_u}$ und $\hat{c}_i \in \mathbb{C}^{n_y}$ jeweils für $1 \leq i \leq q$, die abgeschlossen bzgl. Konjugation sind.

- 1: *Initialisierung:*
- 2: $V = [(\sigma_1 E - A)^{-1} B \hat{b}_1, \dots, (\sigma_1 E - A)^{-1} B \hat{b}_q]$
- 3: $W = [(\sigma_1 E - A)^{-T} C^T \hat{c}_1, \dots, (\sigma_1 E - A)^{-T} C^T \hat{c}_q]$
- 4: $\sigma_{old} = 0 \cdot \sigma$
- 5: *Iteration:*
- 6: **while** $\text{dist}(\sigma_{old}, \sigma) > \varepsilon_I$ **do**
- 7: $E_r = W^T E V$, $A_r = W^T A V$, $B_r = W^T B$, $C_r = C V$
- 8: $\Lambda = \text{diag}(\lambda_i) = \text{eig}(E_r, A_r)$
- 9: Linkseigenvektoren Υ mit $\Upsilon^* A_r = \Lambda \Upsilon^* E_r$
- 10: Rechteigenvektoren Ψ mit $A_r \Psi = E_r \Psi \Lambda$
- 11: normiere, sodass $\Upsilon^* A_r \Psi = \text{diag}(\lambda_i)$ und $\Upsilon^* E_r \Psi = I_r$
- 12: $\sigma_{old} = \sigma$
- 13: setze $\sigma_i = -\lambda_i$, $\hat{b}_i^T = e_i^T \Upsilon^* B_r$ und $\hat{c}_i = C_r \Psi e_i$
- 14: $V = [(\sigma_1 E - A)^{-1} B \hat{b}_1, \dots, (\sigma_1 E - A)^{-1} B \hat{b}_q]$
- 15: $W = [(\sigma_1 E - A)^{-T} C^T \hat{c}_1, \dots, (\sigma_1 E - A)^{-T} C^T \hat{c}_q]$
- 16: **end while**

Output: V, W bzw. $E_r = W^T E V$, $A_r = W^T A V$, $B_r = W^T B$, $C_r = C V$

vorher festgelegt werden muss, beeinflussen die Abbruchschranke ε sowie die Anfangswerte der Interpolationspunkte und Tangentialvektoren das Verhalten des Algorithmus. Ein großer Teil kann jedoch automatisiert bestimmt werden. Zudem besitzt der IRKA – aufgrund der Iteration – eine hohe numerische Robustheit und liefert bei plausiblen Eingabedaten gute Ergebnisse. Eine Konvergenz konnte bislang für den speziellen Fall der Reduktion von „state-space symmetric systems“² bewiesen werden, siehe [FBG12]. Es sind auch Einzelfälle dokumentiert, bei denen der IRKA nicht konvergiert, siehe z.B. [GAB08]. Generell zeigen die vielen sehr guten Reduktionsergebnisse an unterschiedlichen Anwendungsbeispielen die Leistungsfähigkeit des IRKA und lassen weitere Konvergenzbeweise für die Zukunft vermuten.

In dieser Arbeit werden zwei verschiedene Verfahren zur automatisierten Berechnung der Startwerte verwendet: eine am System orientierte zufällige Wahl und eine auf den dominanten Eigenwerten des Originalsystems basierende Wahl. Beide Varianten beruhen auf Ausführungen in [GAB08] für Eingrößensysteme und wurden ausdetailliert und auf den Mehrgrößenfall erweitert. Im Folgenden werden beide Varianten kurz beschrieben.

²Es handelt sich hierbei um lineare Eingrößensysteme mit $A = A^T$ und $B = C$.

Für die zufällige Auswahl nutzt man aus, dass die Interpolationspunkte im gespiegelten Spektrum des reduzierten Systems liegen müssen, damit die Optimalitätsbedingungen erfüllt sind. Es ist zudem zu erwarten, dass sich die Eigenwerte des reduzierten Systems einem Teil der Eigenwerte des Originalsystems annähern. Unter dieser Annahme liegen die Interpolationspunkte auch im gespiegelten Spektrum des Originalsystems. Die Idee zur Bestimmung der Startwerte besteht daher darin, die konvexe Hülle des Spektrums zu bestimmen und die Interpolationspunkte zufällig innerhalb dieses Bereichs zu platzieren. In den in dieser Arbeit betrachteten Anwendungen kann das gesamte Spektrum des Originalsystems berechnet werden, worauf die folgenden Ausführungen beruhen. Bei Systemen mit einer extrem hohen Ordnung ist dies nicht mehr möglich. Für derartige Systeme existieren aber Algorithmen, mit denen das Spektrum abgeschätzt werden kann, indem beispielsweise nur die größten Eigenwerte berechnet werden, siehe erneut [GAB08].

Zur Platzierung der Interpolationspunkte wird zunächst die konvexe Hülle der Eigenwerte $\{\lambda_i\}_{i=1}^{n_x}$ des Originalsystems innerhalb der komplexen Ebene bestimmt. Diese stellt eine Teilmenge

$$K = \{\lambda_{K_j}\}_{j=1}^h \subseteq \{\lambda_i\}_{i=1}^{n_x}$$

der Eigenwerte des Systems dar, bestehend aus h paarweise verschiedenen Eigenwerten. Nun wird $\xi \in [0, 1]^h$ zufällig gewählt und mit Hilfe der Normierung

$$\xi = \frac{1}{\sum_j \xi_j} \cdot \xi$$

sichergestellt, dass $\sum \xi_j = 1$ gilt. Der Interpolationspunkt ip ergibt sich dann als Linearkombination der Eigenwerte aus K zu

$$ip = \xi^T \cdot [\lambda_{K_1}, \dots, \lambda_{K_h}].$$

Auf diese Weise bestimmt man die Hälfte der gewünschten Interpolationspunkte und fügt anschließend die konjugiert komplexen Punkte \bar{ip} hinzu. Im Fall einer ungeraden Ordnung q des reduzierten Modells wird abschließend ein reeller Interpolationspunkt

$$ip = \xi^T \cdot \Re([\lambda_{K_1}, \dots, \lambda_{K_h}]).$$

hinzugefügt. Die Tangentialvektoren \hat{c}_i und \hat{b}_i^T werden vollständig durch Zufallszahlen festgelegt. Eine beispielhafte Anwendung dieser Variante zur Bestimmung der Startwerte für den IRKA befindet sich im nachfolgenden Abschnitt 2.5.

Die zweite im Rahmen dieser Arbeit implementierte Methode zur Generierung von Startwerten basiert auf dem modalen Abschneiden, einem Verfahren zur

MOR, bei dem bestimmte Eigenwerte des Originalsystems beibehalten und die übrigen abgeschnitten werden. Dies wird erreicht, indem die Links- und Rechts-eigenvektoren der zu behaltenden Eigenwerte als Projektionsmatrizen verwendet werden, siehe z.B. [SVR08]. Das reduzierte System liegt dann in Diagonalform

$$\underbrace{I_q}_{W^T E V} \dot{x}_r = \underbrace{\begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_q \end{bmatrix}}_{W^T A V} x_r + \underbrace{\hat{B}}_{W^T B} u,$$

$$y = \underbrace{\hat{C}}_{C V} x_r$$

vor. Für den IRKA ist die Form der Übertragungsfunktion von entscheidender Bedeutung. Sie lässt sich ausdrücken als

$$G_r(s) = \sum_{i=1}^q \frac{\hat{c}_i \hat{b}_i^T}{s - \lambda_i}$$

und besitzt damit die in (2-12) verwendete Darstellung. Es kann zwar keineswegs garantiert werden, dass das mittels des modalen Abschneidens berechnete reduzierte System bereits optimal im Sinne der \mathcal{H}_2 -Norm ist. Es ist aber dennoch sinnvoll die dominanten Eigenwerte λ_i gespiegelt an der imaginären Achse als Interpolationspunkte zu verwenden. Darüber hinaus hat man zusätzlich mit \hat{c}_i und \hat{b}_i^T zum System „passende“ Tangentialvektoren zur Verfügung. Basierend hierauf ist es naheliegend, ein verbessertes Konvergenzverhalten des IRKA durch diese Wahl der Anfangswerte zu vermuten. Das im nächsten Abschnitt behandelte Beispiel bestätigt dies jedoch nur zum Teil. Zur Bestimmung der dominanten Eigenwerte wird das sogenannte Litzsche Dominanzmaß verwendet, das z.B. in [Föl08] oder [Lit79] beschrieben ist.

Zusammengefasst hängt der IRKA unter Berücksichtigung dieser beiden Varianten der Startwertberechnung nur noch von zwei Steuerungsparametern ab: der Schranke ε und der Ordnung q des reduzierten Modells. Erstere besitzt keinen großen Einfluss auf die Approximationsgüte, solange sie moderate Werte annimmt. Daher ist die MOR mit dem IRKA fast vollständig automatisierbar, beispielsweise indem die Ordnung q sukzessive erhöht wird bis eine vorgegebene Abweichung im Sinne der \mathcal{H}_2 - oder \mathcal{H}_∞ -Norm unterschritten wird.

2.5 Anwendung am Beispiel des Aktormoduls

In diesem Abschnitt werden einige Ergebnisse der innerhalb dieses Kapitels vorgestellten Algorithmen dargestellt. Als Anwendungsbeispiel dient das linke Aktormodul des Feder-Neige-Prüfstands, siehe Abschnitt 1.3. Mit dem Aktormodul

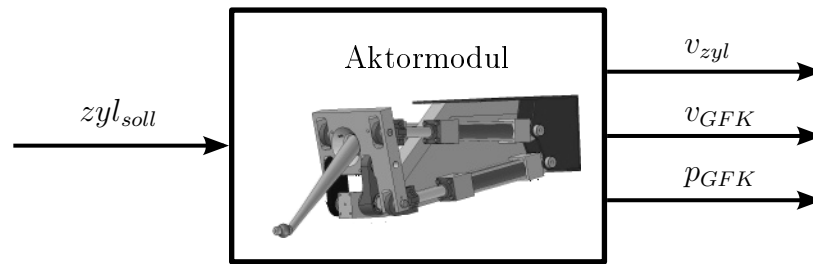


Bild 2-2: Blockbild des Aktormoduls mit den verwendeten Ein- und Ausgängen

wird durch eine aktive Verstellung des Federfußpunktes eine Kraft auf den Aufbau erzeugt. Es lässt sich im Rahmen der hierarchischen Modellierung als unterlagertes System separat vom Gesamtsystem betrachten.

Es handelt sich bei dem betrachteten Modell, dargestellt in Bild 2-2, um ein lineares Modell der Ordnung 18 mit drei Eingängen und sieben Ausgängen. Die Eingänge stellen die drei Sollwerte der Zylinderpositionen zyl_{soll} dar, die innerhalb des Gesamtsystems über eine inverse Kinematik vorgegeben werden. Die Ausgänge werden zum einen von den drei Zylindergeschwindigkeiten v_{zyl} gebildet, die zur Abschätzung der hydraulischen Leistung verwendet werden können. Zum anderen werden die Positionen p_{GFK} und Geschwindigkeiten v_{GFK} des Federfußpunktes, jeweils in vertikaler und horizontaler Richtung, benötigt, um innerhalb des Gesamtsystems die in der GFK-Feder wirkende Kraft zu berechnen. Die Rückwirkung der Kraft auf das Aktormodul selbst kann aufgrund der unterlagerten Zylinderregler vernachlässigt werden.

Das Modell des Aktormoduls wird zunächst auf Ordnung 9 reduziert. Dazu werden sowohl die manuelle rationale Interpolation aus Abschnitt 2.3, als auch der IRKA mit den beiden Startwertvarianten aus Abschnitt 2.4 verwendet.

Die manuelle rationale Interpolation führt zu guten Ergebnissen, wenn die drei Entwicklungspunkte $\{-10, -20, -100\}$ verwendet werden. Zu jedem Entwicklungspunkt werden drei Basisvektoren der zugehörigen Krylov-Unterräume berechnet. Diese Auswahl wurde empirisch getroffen und gestaltete sich relativ schwierig, da selbst sehr ähnliche Kombinationen von Entwicklungspunkten und Basisvektoren zu instabilen reduzierten Systemen führen.

Für den IRKA wurden, wie bereits erwähnt, beide Startwertvarianten genutzt. In Bild 2-3 sind die Interpolationspunkte dargestellt. Die konvexe Hülle des Spektrums des Aktormoduls wurde an der imaginären Achse gespiegelt und ist in blau eingezeichnet. Die zufällig ausgewählten Interpolationspunkte befinden sich in etwa mittig innerhalb des gespiegelten Spektrums. Die modalen Interpolationspunkte befinden sich am linken Rand. Es handelt sich bei ihnen um die 9 Eigenwerte mit dem größten Dominanzmaß. In beiden Fällen ist der IRKA gegen

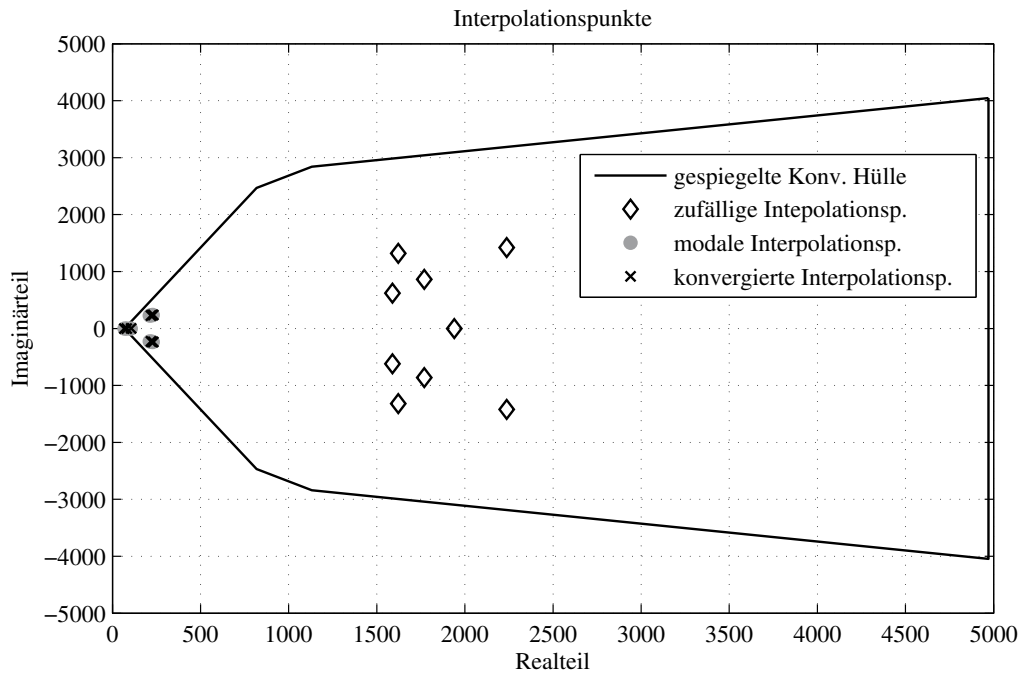


Bild 2-3: Interpolationspunkte für die Tangentiale Interpolation vor bzw. nach der Anwendung des IRKA

die gleichen Interpolationspunkte konvergiert und lieferte ein stabiles reduziertes System. Die konvergierten Interpolationspunkte sind ebenfalls in Bild 2-3 zu finden. Sie unterscheiden sich für diesen konkreten Fall nur unwesentlich von den modalen Startwerten. Dies spiegelt sich auch in der Anzahl der Iterationen wieder. Bei der Verwendung der modalen Startwerte konvergierte der IRKA bereits nach 3 Schritten. Bei den zufälligen Startwerten wurden 21 Iterationen benötigt. Diese Zahl variierte zudem stark bei einer mehrfachen Ausführung mit jeweils anderen zufällig gewählten Werten.

In Bild 2-4 ist beispielhaft die Sprungantwort vom Eingang $zyl_{1,soll}$ zum Ausgang y_{GFK} für alle drei reduzierten Systeme, sowie die des Originalsystems dargestellt. Die hohe Approximationsgüte aller reduzierten Modelle wird deutlich. Optisch sind in der gewählten Darstellung keinerlei Unterschiede erkennbar. Die übrigen Ein- / Ausgangskombinationen weisen ein vergleichbar gutes Ergebnis auf und werden daher an dieser Stelle nicht näher betrachtet.

Unterschiede zwischen der manuellen rationalen Interpolation und der tangentialen Interpolation mittels des IRKA werden über die Reduktionsfehler deutlich. In Tabelle 2-1 sind die relativen Abweichungen zwischen dem Originalsystem und den jeweiligen reduzierten Systemen eingetragen. Da der IRKA für beide Initia-

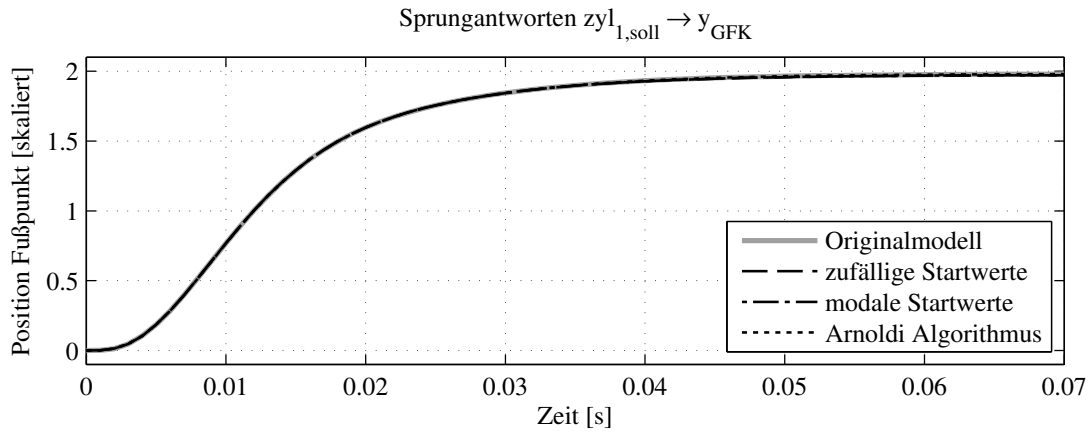


Bild 2-4: Vergleich der Sprungantworten des Aktormoduls nach einer Reduktion mit dem Arnoldi-Algorithmus und den beiden IRKA-Varianten

lisierungsvarianten zu identischen Interpolationspunkten konvergiert ist, unterscheiden sich diese Varianten nicht. Es ist jedoch zu erkennen, dass sowohl bzgl. der \mathcal{H}_2 - als auch der \mathcal{H}_∞ -Norm die Abweichungen bei der manuellen rationalen Interpolation mit dem Arnoldi-Algorithmus größer sind.

Interessant ist an dieser Stelle, dass die stationäre Abweichung zwischen Original- und reduziertem System jedoch bei der manuellen rationalen Interpolation deutlich geringer ist. Dies ist auf die Lage der Entwicklungspunkte zurückzuführen. Die manuell gewählten Entwicklungspunkte liegen deutlich näher am Ursprung, als die Interpolationspunkte beim IRKA. Da ein Momentenabgleich an der Frequenz $s = 0$ die stationäre Genauigkeit sicherstellt, ist für Entwicklungspunkte in der Nähe des Ursprungs ebenfalls ein kleiner stationärer Fehler zu erwarten.

Aus dem hier vorgestellten Anwendungsbeispiel wird deutlich, dass beide Algorithmen zu guten Ergebnissen führen können. Das Beispiel der stationären Abweichung unterstreicht den Vorteil der manuellen Interpolation, bestimmte Aspekte des Originalsystems im reduzierten System besser berücksichtigen zu können. Allerdings besitzt die manuelle Interpolation den großen Nachteil, dass die Suche

Tabelle 2-1: Relative Fehler bei der Reduktion des Aktormoduls

	IRKA (zufällig)	IRKA (modal)	Arnoldi
\mathcal{H}_2 -Fehler	$5,7017 \cdot 10^{-2}$	$5,7017 \cdot 10^{-2}$	$6,1311 \cdot 10^{-2}$
\mathcal{H}_∞ -Fehler	$2,2799 \cdot 10^{-2}$	$2,2799 \cdot 10^{-2}$	$2,4845 \cdot 10^{-2}$
stat. Fehler	$1,241 \cdot 10^{-1}$	$1,241 \cdot 10^{-1}$	$1,0783 \cdot 10^{-5}$

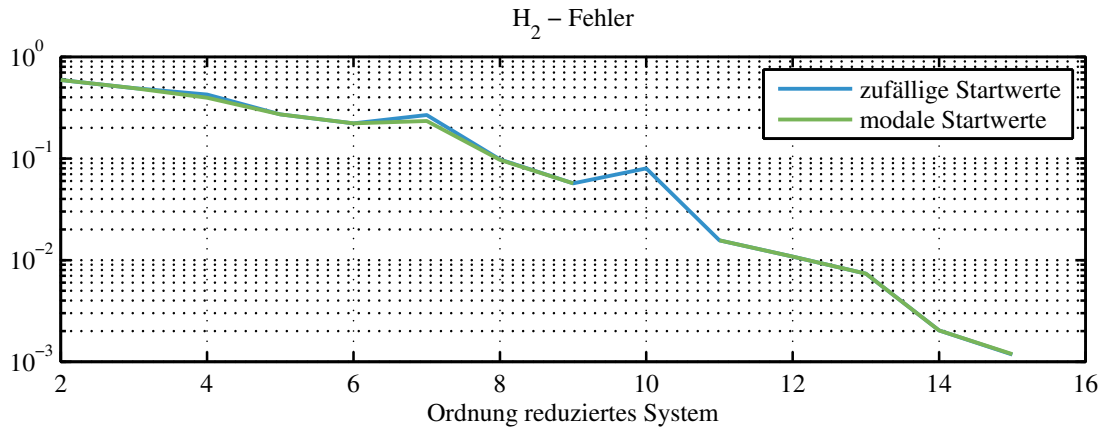


Bild 2-5: Relativer \mathcal{H}_2 -Fehler für unterschiedliche Ordnungen der reduzierten Systeme

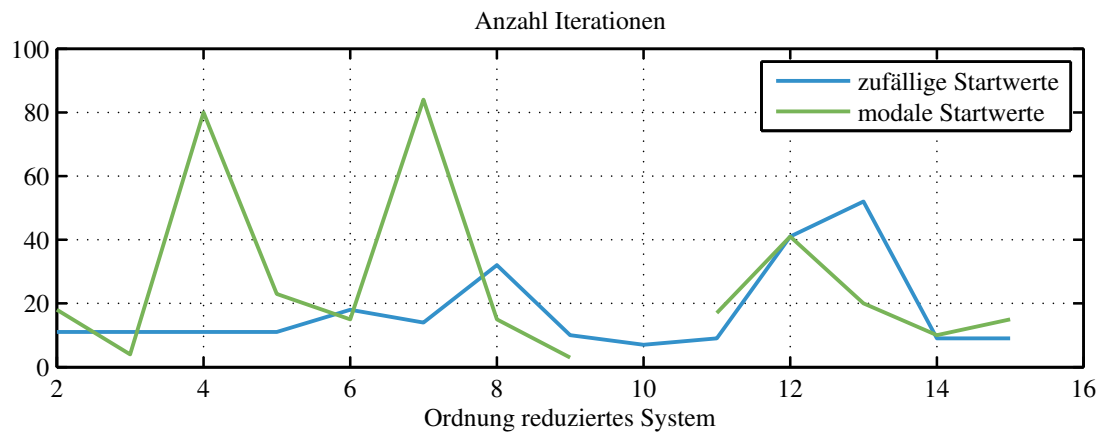


Bild 2-6: Benötigte Iterationen innerhalb des IRKA bis zur Konvergenz

nach geeigneten Entwicklungspunkten aufwändig sein kann. Der IRKA liefert hingegen eine sehr gute Approximation der reduzierten Modelle selbst bei zufällig festgelegten Startwerten.

Dieser Vorteil der leichteren Automatisierbarkeit wird besonders deutlich, wenn man reduzierte Systeme unterschiedlicher Ordnungen miteinander vergleichen möchte. Für das Aktormodul war es insbesondere für kleine Ordnungen nicht möglich mit vertretbarem Aufwand stabile reduzierte Systeme bei einer manuellen Interpolation zu finden. Für den IRKA war dies problemlos möglich.

In Bild 2-5 sind die relativen Fehler bzgl. der \mathcal{H}_2 -Norm für unterschiedliche Ordnungen aufgetragen. Bild 2-6 zeigt die zugehörigen Iterationen bis die Interpo-

Tabelle 2-2: Innerhalb dieser Arbeit verwendete Verfahren der MOR

Rationale Interpolation der Übertragungsfunktion		
Statisches Originalsystem		
	<u>manuelle Interpolation</u>	<u>\mathcal{H}_2-optimale tangentielle Interpolation</u>
Algorithmus:	Arnoldi-Algorithmus	Iterative Rational Krylov Algorithm (IRKA)
Parameter:	<ul style="list-style-type: none"> • Deflationsschranke ε_D • Entwicklungspunkte s_i • Anzahl Basisvektoren q_i 	<ul style="list-style-type: none"> • Abbruchschranke ε_I • Init. Entwicklungspunkte σ_i • Init. Tangentialvektoren \hat{b}_i, \hat{c}_i
Eigenschaften:	<ul style="list-style-type: none"> ✓ Einfluss auf Approximationseigenschaften (z.B. stat. Genauigkeit) ✗ aufwändige manuelle Auswahl von s_i und q_i 	<ul style="list-style-type: none"> ✓ notwendige Bedingung für Optimalität im Konvergenzfall ✓ Initialisierung automatisierbar ✗ zumeist keine stationäre Genauigkeit des reduzierten Modells

lationspunkte konvergiert sind. Es wurden erneut beide Startwertvarianten, die zufälligen und die modalen Startwerte, genutzt.

Auffällig ist hier, dass besonders bei den modalen Anfangswerten die Anzahl der Iterationen sehr stark variiert. Für die Ordnung 10 konnte keine Konvergenz erzielt werden. Das zugehörige red. System, basierend auf den Interpolationspunkten im letzten Iterationsschritt, ist sogar instabil, weshalb an dieser Stelle kein endlicher \mathcal{H}_2 -Fehler vorliegt.

Es ist dann umso interessanter, dass dennoch für die Ordnung 10 ein brauchbares reduziertes System basierend auf zufälligen Startwerten berechnet werden kann und der IRKA sogar nur 7 Iterationen benötigt. Allerdings ist auch hier anzumerken, dass insbesondere die Anzahl der Iterationen, aber auch der relative \mathcal{H}_2 -Fehler von der konkreten Lage der initialen Interpolationspunkte abhängen. Es traten hier durchaus auch bei zufälligen Interpolationspunkten instabile Systeme sowie Konvergenzprobleme auf.

Zusammengefasst sprechen die Ergebnisse des Anwendungsbeispiels dafür, dass sowohl die manuelle rationale Interpolation, als auch die \mathcal{H}_2 -optimale Interpolation für mechatronische Systeme gut geeignet sind. In Tabelle 2-2 sind die wichtigs-

ten Eigenschaften einander gegenüber gestellt. Hinsichtlich der Automatisierbarkeit ist ganz klar der IRKA als Algorithmus zu bevorzugen. Allerdings muss auch dann das Reduktionsergebnis überprüft werden, da Konvergenzprobleme und instabile Systeme möglich sind. Eine einfache pragmatische Lösung besteht, neben einer Erhöhung der Ordnung, jedoch meist in der wiederholten Ausführung mit zufälligen Startwerten bis brauchbare reduzierte Systeme vorliegen. Die manuelle rationale Interpolation führt zu einer vergleichbaren Approximationsgüte mit dem Vorteil, dass je nach gegebenen Anforderungen auch spezielle Eigenschaften wie die stationäre Genauigkeit beeinflusst werden können. Die Auswahl geeigneter Parameter für den Arnoldi-Algorithmus ist jedoch oftmals zeitaufwändig und kaum automatisierbar.

3 Parametrische Modellordnungsreduktion

Ausgehend von den Verfahren zur Modellordnungsreduktion, die innerhalb des vorhergehenden Kapitels vorgestellt wurden, beschäftigt sich dieses Kapitel mit den im Rahmen dieser Arbeit eingesetzten Methoden der parametrischen Modellordnungsreduktion (PMOR). Es werden zwei unterschiedliche Verfahren zunächst allgemein beschrieben. Eine Anwendung der Verfahren im Kontext selbstoptimierender Systeme erfolgt im Rahmen von Kapitel 5.

Bei den Verfahren handelt es sich zum einen um die manuelle rationale Interpolation parametrischer Systeme in Abschnitt 3.1 und zum anderen um die Matrix Interpolation in Abschnitt 3.2. Für das erste Verfahren wurde innerhalb dieser Arbeit ein objektorientierter Arnoldi-Algorithmus entwickelt. Daher besteht dieser Abschnitt aus zwei Teilen. Zuerst werden die theoretischen Grundlagen behandelt und nachfolgend wird der neuartige Arnoldi-Algorithmus beschrieben. Die zweite Methode basiert auf einer geschickten Kombination der Modellordnungsreduktion nichtparametrischer Systeme und benötigt keinen speziellen Algorithmus. Die Darstellung in Abschnitt 3.2 beschränkt sich daher auf die mathematischen Zusammenhänge.

3.1 Manuelle rationale Interpolation parametrischer Systeme

Das in diesem Abschnitt vorgestellte Verfahren zur parametrischen MOR stellt eine direkte Erweiterung der manuellen rationalen Interpolation aus Abschnitt 2.3 dar. Das Ziel, die Momente der Übertragungsfunktion abzugleichen, wird auf Systeme erweitert, die von zusätzlichen Parametern abhängen. Diese Parameterabhängigkeit soll auch im reduzierten System erhalten bleiben. In ersten Arbeiten hierzu wurde zunächst ein einziger zusätzlicher Parameter betrachtet und nur einseitige Projektionen berechnet, siehe [WMGG99] und [WM01]. Die grundlegenden theoretischen Resultate für eine beliebige Anzahl an Parametern gehen auf [DSC⁺04] zurück. Der dort beschriebene Momentenabgleich ist auf Systeme anwendbar, die eine affin-lineare Parameterabhängigkeit besitzen. Allerdings werden bereits Strategien präsentiert, wie über die Einführung zusätzlicher Parameter auch allgemeine, nichtlineare Parameterabhängigkeiten behandelt werden können. Der Artikel [Fen05] gibt einen Überblick über verschiedene Ausprägungen des mehrdimensionalen Momentenabgleichs und enthält Vergleiche hinsichtlich der algorithmischen Komplexität der unterschiedlichen Varianten.

In der darauffolgenden Zeit entstanden eine Reihe von Publikationen mit Anwendungen des mehrdimensionalen Momentenabgleichs. Beispielsweise werden in [FHIDE06] Finite-Elemente Modelle mit polynomieller Parametrierung behandelt oder in [RMG⁺06] Modelle von Schaltkreises unter Beibehaltung bestimmter Entwurfsparameter reduziert. Auch mechanische Systeme, beschrieben durch DGL-Systeme 2. Ordnung, wurden untersucht, siehe [ESL⁺06].

Der rekursive Ansatz, der innerhalb dieser Arbeit verfolgt und implementiert wird, basiert auf Resultaten aus [FB07], die wesentlich zur Vereinfachung der theoretischen Zusammenhänge beigetragen haben. Ähnliche Beobachtungen, allerdings beschränkt auf Eingrößensysteme, werden in [LBS09] ausgenutzt. Das Ergebnis ist ein spezieller Arnoldi-Algorithmus mit variierenden Krylov-Unterräumen, bezeichnet als „two-directional Arnoldi Process“.

Es lassen sich nur noch wenige neue Publikationen aus dem Bereich des mehrdimensionalen Momentenabgleichs finden. Dies liegt sicherlich zum Teil an den Nachteilen dieses Ansatzes. Die Freiheitsgrade zur Berechnung des reduzierten Systems müssen aufwändig manuell festgelegt werden und bei einer größeren Anzahl zu erhaltender Parameter steigt tendenziell die Ordnung der reduzierten Systeme an. Weitere Details hierzu sind am Ende dieses Abschnitts zu finden. Einer der Vorteile des mehrdimensionalen Momentenabgleichs besteht darin, dass auch Ableitungen nach den Parametern intuitiv berücksichtigt werden können. Die Arbeit [YM12] greift dies auf und nutzt die Ableitungen nach den Parametern, um die Optimierung mit Hilfe von Quasi-Newton Verfahren zu beschleunigen. Die Suchrichtung innerhalb der Optimierung wird hierbei als Reduktionsparameter der Systeme verwendet. Durch die enge Verzahnung eines Optimierungsverfahrens mit der PMOR entsteht eine gewisse Ähnlichkeit mit der im Rahmen der vorliegenden Arbeit durchgeführten hierarchischen Optimierung basierend auf parametrischen reduzierten Systemen.

Der Rest dieses Abschnittes ist in zwei Teile unterteilt. Zunächst erfolgt eine Darstellung der theoretischen Grundlagen des mehrdimensionalen Momentenabgleichs im Unterabschnitt 3.1.1. Anschließend wird in 3.1.2 der im Rahmen dieser Arbeit entwickelte Algorithmus zur Berechnung der Projektionsmatrizen vorgestellt.

3.1.1 Theoretische Grundlagen

Es handelt sich bei der manuellen rationalen Interpolation parametrischer Systeme um eine Verallgemeinerung der manuellen rationalen Interpolation auf parameterabhängige Systeme der Form

$$\begin{aligned} \dot{x} &= A(p)x + Bu, \\ y &= Cx \end{aligned} \tag{3-1}$$

mit dem Parametervektor $p = [p_1 \ \dots \ p_{n_p}]^T \in \mathbb{R}^{n_p}$. Prinzipiell ist es ebenfalls möglich, parameterabhängige Eingangs- bzw. Ausgangsmatrizen zu verwenden, die folgenden Herleitungen werden dadurch aber deutlich umfangreicher. Da zudem bei den in dieser Arbeit betrachteten Anwendungsfällen nur konstante Ein- und Ausgangsmatrizen vorliegen, werden an dieser Stelle nur Systeme der obigen Form behandelt. Eine weitere Einschränkung ist an die Art der Parameterabhängigkeit von A zu stellen. Für die Gültigkeit der folgenden Ausführungen ist es erforderlich, dass die Systemmatrix über eine affin-lineare Parameterabhängigkeit verfügt, d. h.

$$A(p) = A_0 + p_1 A_1 + p_2 A_2 + \dots + p_{n_p} A_{n_p}.$$

Im Allgemeinen wird diese Form nicht gegeben sein. Die Parameterabhängigkeit kann sogar unbekannt sein, falls das Modell mit Hilfe von rechnergestützten Modellierungswerkzeugen erstellt wurde. In derartigen Fällen kann die parametrische Reduktion angewendet werden, indem die affin-lineare Darstellung mittels einer Taylorentwicklung der Systemmatrix nach den Parametern erzeugt wird. Man erhält dann allerdings nur eine lokale Näherung, deren Gültigkeitsbereich für den konkreten Anwendungsfall untersucht werden muss. Weitere Details hierzu sowie die Erweiterung auf polynomielle Parameterabhängigkeiten sind in [DSC⁺04] zu finden.

Die Übertragungsfunktion von (3-1) ist neben der Laplace-Variablen s nun ebenfalls von den Parametern p abhängig und besitzt die Form

$$G(s, p) = \frac{Y(s)}{U(s)} = C (sI - A_0 - p_1 A_1 - p_2 A_2 - \dots - p_{n_p} A_{n_p})^{-1} B.$$

Führt man nun analog zu den nichtparametrischen Betrachtungen des vorhergehenden Kapitels den Entwicklungspunkt s_0 und die Hilfsvariable $\sigma := s - s_0$ ein, so erhält man

$$G(\sigma, p) = C (-(A_0 - s_0 I) + \sigma I - p_1 A_1 - \dots - p_{n_p} A_{n_p})^{-1} B.$$

Um die Momente zu erhalten, wird die Übertragungsfunktion erneut in eine Reihendarstellung überführt. Zur Vereinfachung der Notation werden die nachfolgend definierten Abkürzungen M_0, \dots, M_{n_p} verwendet. Zudem werden sämtliche Parameter in einem Vektor

$$\tilde{p} = (p_0 \ p_1 \ p_2 \ \dots \ p_{n_p})^T := (\sigma \ p_1 \ p_2 \ \dots \ p_{n_p})^T \in \mathbb{R}^{n_p+1}$$

zusammengefasst. Man erhält dann zunächst

$$G(p) = C \left[I - \underbrace{((A_0 - s_0 I)^{-1})}_{M_0} \underbrace{\sigma}_{p_0} \underbrace{-(A_0 - s_0 I)^{-1} A_1}_{M_1} p_1 - \dots \right. \\ \left. \underbrace{-(A_0 - s_0 I)^{-1} A_{n_p}}_{M_{n_p}} p_{n_p} \right]^{-1} \underbrace{(-(A_0 - s_0 I)^{-1} B)}_{B_M}. \quad (3-2)$$

Nimmt man erneut an, dass die Neumannsche Reihe

$$\sum_{j=0}^{\infty} (M_0 p_0 + M_1 p_1 + \dots + M_{n_p} p_{n_p})^j$$

existiert und somit

$$(I - (M_0 p_0 + M_1 p_1 + \dots + M_{n_p} p_{n_p}))^{-1} = \sum_{j=0}^{\infty} (M_0 p_0 + M_1 p_1 + \dots + M_{n_p} p_{n_p})^j$$

gilt, so lässt sich die Übertragungsfunktion als unendliche Summe schreiben:

$$G(p) = C \sum_{j=0}^{\infty} (M_0 p_0 + M_1 p_1 + \dots + M_{n_p} p_{n_p})^j B_M. \quad (3-3)$$

Analog zu den Betrachtungen in Kapitel 2 werden die Koeffizienten dieser Reihe als Momente bezeichnet. Aufgrund der Abhängigkeit von $(n_p + 1)$ Parametern statt eines einzigen, existieren nun jedoch $(n_p + 1)^k$ Momente der Ordnung k (siehe auch Bild 3-1):

- 0. Ordnung: CB_M
- 1. Ordnung: $CM_i B_M \quad i = 0, \dots, n_p$
- 2. Ordnung: $CM_{i_1} M_{i_2} B_M, \quad i_1, i_2 = 0, \dots, n_p$
- \vdots
- k. Ordnung: $CM_{i_1} \dots M_{i_k} B_M, \quad i_1, \dots, i_k = 0, \dots, n_p$

Die Reduktion selbst wird wie im nichtparametrischen Fall durch die Projektion mit den beiden Projektionsmatrizen V und W durchgeführt. Wählt man

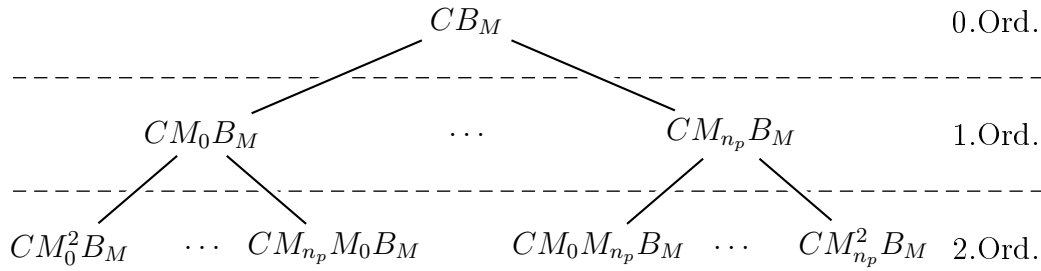


Bild 3-1: Momente der Übertragungsfunktion eines parametrischen Systems

die Projektionsmatrix V als Basis des parametrischen, eingangsseitigen Krylov-Unterraums

$$\mathcal{K}([M_0, \dots, M_{n_p}], B_M) = [B_M, M_0B_M, \dots, M_{n_p}B_M, M_0^2B_M, M_0M_1B_M, \dots], \quad (3-4)$$

so stimmen abhängig von der Anzahl der Basisvektoren die ersten Momente überein. Das reduzierte Modell ist dabei gegeben durch

$$\begin{aligned} E_r \dot{x}_r &= A_r(p)x + B_ru, \\ y_r &= C_rx_r \end{aligned} \quad (3-5)$$

mit

$$\begin{aligned} A_r(p) &= A_{0r} + p_1A_{1r} + \dots + p_{n_p}A_{n_p r} \\ &= W^T A(\tilde{p})V = W^T A_0V + p_1W^T A_1V + \dots + p_{n_p}W^T A_{n_p}V, \\ E_r &= W^T V, \quad B_r = W^T B \quad \text{und} \quad C_r = CV. \end{aligned}$$

Die zweite Projektionsmatrix W ist bei einseitigen Verfahren beliebig wählbar, mit der aus der nichtparametrischen Reduktion bekannten Einschränkung, dass $W^T V$ invertierbar sein muss.

In [FB07] wird zur stabilen numerischen Berechnung der Basis V eine rekursive Formulierung des Unterraumes (3-4) eingeführt, die sich aus der Reihendarstellung (3-3) ergibt. Die Matrix V kann demnach alternativ zu (3-4) als Basis des Unterraumes

$$[R_0, R_1, R_2, \dots]$$

mit

$$\begin{aligned} R_0 &= B_M \in \mathbb{R}^{n \times p}, \\ R_1 &= [M_0 R_0, M_1 R_0, \dots, M_{n_p} R_0] \in \mathbb{R}^{n \times (n_p+1)p}, \\ R_2 &= [M_0 R_1, M_1 R_1, \dots, M_{n_p} R_1] \in \mathbb{R}^{n \times (n_p+1)^2 p}, \\ &\vdots \\ R_k &= [M_0 R_{k-1}, M_1 R_{k-1}, \dots, M_{n_p} R_{k-1}] \in \mathbb{R}^{n \times (n_p+1)^k p}, \end{aligned}$$

gewählt werden.

Auch beim mehrdimensionalen Momentenabgleich kann das einseitige Verfahren auf ein zweiseitiges erweitert werden, um die Anzahl der abgeglichenen Momente zu erhöhen. Zur Berechnung der zweiten Projektionsmatrix W wird diese, ebenfalls analog zu den Überlegungen in Kapitel 2, als Basis eines weiteren Krylov-Unterraumes gewählt, der im Folgenden hergeleitet wird.

Den Ausgangspunkt bildet erneut die in eine Reihe entwickelte Übertragungsfunktion

$$G(p) = C \sum_{j=0}^{\infty} (M_0 p_0 + M_1 p_1 + \dots + M_{n_p} p_{n_p})^j (-(A_0 - s_0 I)^{-1} B).$$

Definiert man nun zunächst

$$C_M := A_0^{-T} C^T,$$

so lässt sich analog zum Unterraum (3-4) ein Unterraum für die ausgangsseitigen Beziehungen aufstellen. Dazu sei zur weiteren Vereinfachung der Notation die folgende Schreibweise definiert: Sind $\mathcal{A}_i \in \mathbb{R}^{n \times n}$, $i = 1, \dots, m$ beliebige quadratische Matrizen, so bezeichnet

$$(\mathcal{A}_1 \mathcal{A}_2 \cdot \dots \cdot \mathcal{A}_m)^\tau := \mathcal{A}_1^T \cdot \mathcal{A}_2^T \cdot \dots \cdot \mathcal{A}_m^T$$

das einzelne Transponieren der Matrizen, ohne die Reihenfolge der Multiplikation zu vertauschen. Mit dieser Notation ist der ausgangsseitige parametrische Krylov-Unterraum gegeben als:

$$\mathcal{K}([M_0^\tau, \dots, M_{n_p}^\tau], C_M) = [C_M, M_0^\tau C_M, \dots, M_{n_p}^\tau C_M, (M_0^\tau)^2 C_M, M_0^\tau M_1^\tau C_M, \dots]. \quad (3-6)$$

Auch hier lässt sich eine rekursive Darstellung

$$\mathcal{K}([M_0^\tau, \dots, M_{n_p}^\tau], C_M) = [R_0, R_1, R_2, \dots]$$

gewinnen, indem man folgende Definitionen trifft (siehe erneut [FB07]):

$$\begin{aligned}
R_0 &= C_M \in \mathbb{R}^{n \times p}, \\
R_1 &= \begin{bmatrix} M_0^\tau R_0, & M_1^\tau R_0, & \dots, & M_{n_p}^\tau R_0 \end{bmatrix} \in \mathbb{R}^{n \times (n_p+1)p}, \\
R_2 &= \begin{bmatrix} M_0^\tau R_1, & M_1^\tau R_1, & \dots, & M_{n_p}^\tau R_1 \end{bmatrix} \in \mathbb{R}^{n \times (n_p+1)^2 p}, \\
&\vdots \\
R_k &= \begin{bmatrix} M_0^\tau R_{k-1}, & M_1^\tau R_{k-1}, & \dots, & M_{n_p}^\tau R_{k-1} \end{bmatrix} \in \mathbb{R}^{n \times (n_p+1)^k p}.
\end{aligned}$$

Wird nun W als Basis von $\mathcal{K}([M_0^\tau, \dots, M_{n_p}^\tau], C_M)$ gewählt, so lässt sich zeigen, dass abhängig von der Dimension von W ebenfalls die ersten Momente abgeglichen werden. Bei der Verwendung beider Krylov-Unterräume addiert sich auch für parametrische Systeme die Anzahl der abgeglichenen Momente.

Die effiziente und stabile numerische Berechnung geeigneter Basen der hier vorgestellten Krylov-Unterräume stellt eine Herausforderung an eine algorithmische Umsetzung dar. Neben der im letzten Kapitel geschilderten Problematik mehrere Ein- und Ausgänge zu behandeln, tritt nun noch eine variable Anzahl von Parametern auf, die zu zusätzlichen Momenten führen. Dies stellt einen bedeutenden Nachteil des mehrdimensionalen Momentenabgleichs dar. Je mehr Parameter in der Reduktion erhalten werden sollen, desto mehr Basisvektoren sind erforderlich, um beispielsweise alle Momente 1. Ordnung abzugleichen. Dies hat zur Folge, dass die Ordnung des reduzierten Modells mit der Anzahl der Parameter ansteigt. Dies ist ein Effekt, der manchmal als „curse of dimensionality“ („Fluch der Dimensionalität“) bezeichnet wird.

Im folgenden Unterabschnitt wird zunächst ein effizienter Algorithmus zur Berechnung von V und W vorgestellt. Die Vermeidung des zuletzt geschilderten „curse of dimensionality“ durch eine spezielle Form der Interpolation selbstoptimierender Systeme wird in Kapitel 5 beschrieben und stellt ein wesentliches Resultat dieser Arbeit dar.

3.1.2 Rekursiver objektorientierter Arnoldi-Algorithmus

In diesem Unterabschnitt wird eine neuartige Variante des Arnoldi-Algorithmus vorgestellt, mit der Projektionsmatrizen für die Reduktion parametrischer Systeme berechnet werden können. Die Projektionsmatrizen werden erneut aus Basisvektoren der im letzten Unterabschnitt angegebenen Krylov-Unterräume gebildet. Die Zielsetzung des Algorithmus besteht aus der numerisch stabilen Berechnung dieser Basisvektoren. Dabei müssen eine Reihe von Anforderungen erfüllt sein.

Die Anzahl der Basisvektoren muss frei vorgebar sein, damit reduzierte Systeme beliebiger Ordnung erstellt werden können. Da die Krylov-Unterräume (3-4) und

Algorithmus 3 Objekt-orientierter Arnoldi-Algorithmus für die PMOR

Input: Systemmatrizen A_j , $j = 0, \dots, n_p$, B, C, D ; Ordnung q ; Deflationsschranke ε_D ; Entwicklungspunkt s_0

- 1: Baumstruktur der Klasse *projTree* initialisieren für V (treeV)
- 2: Baumstruktur der Klasse *projTree* initialisieren für W (treeW)
- 3: **for** $i = 1 : q$ **do**
- 4: treeV.addNodeToTree
- 5: treeW.addNodeToTree
- 6: **end for**
- 7: $V = \text{treeV.V}$
- 8: $W = \text{treeW.V}$

Output: Projektionsmatrizen V und W

(3-6) von dem gewählten Entwicklungspunkt s_0 abhängen, muss dieser ebenfalls vom Benutzer festzulegen sein. Insbesondere aufgrund der hohen Anzahl der Momente durch die zusätzlichen Parameter muss die Möglichkeit der Deflation, d. h. des Aussortierens linear abhängiger Basisvektoren, gegeben sein.

Grundsätzlich kann weiterhin ein Arnoldi-Algorithmus verwendet werden. Im nichtparametrischen Fall werden während der Iteration die vorhandenen Basisvektoren mit der quadratischen Matrix A_K multipliziert und daraus neue Basisvektoren berechnet (siehe Algorithmus 1). Im parametrischen Fall sind nun, abhängig von der Anzahl der Parameter, mehrere Matrizen, nämlich M_0, \dots, M_{n_p} , zu verwenden, wie aus der Baumdarstellung in Bild 3-1 deutlich wird.

In [Sch09], [Krü09] sowie in [KST10] wurde ein zweiseitiger Arnoldi-Algorithmus für den mehrdimensionalen Momentenabgleich entwickelt. Die Implementierung orientiert sich sehr eng an der Arnoldi-Grundform und bildet den Ausgangspunkt für den im Folgenden beschriebenen objektorientierten Algorithmus.

Aufgrund der rekursiven Abhängigkeit der Momente voneinander wie sie in Bild 3-1 gezeigt ist, bietet es sich an, die Basisvektoren mit Hilfe objektorientierter Programmierung auf der Grundlage dynamischer Datenstrukturen zu realisieren. Eine derartige Herangehensweise an den mehrdimensionalen Momentenabgleich wurde erstmals im Rahmen der vorliegenden Arbeit umgesetzt.

Den Kern bilden die beiden Klassen *projTree* und *projVector*, siehe Tabelle 3-1 und 3-2. Jede Instanz der Klasse *projTree* repräsentiert eine Projektionsmatrix inklusive aller zur Berechnung erforderlichen Daten. Die einzelnen Komponenten und ihr Zusammenspiel werden nachfolgend erläutert. Zunächst sei aber auf den vom Benutzer zur Reduktion auszuführenden Algorithmus 3 hingewiesen. Dieser ist sehr übersichtlich und verfügt über eine einfache Struktur, was einen der Vorteile der objektorientierten Herangehensweise ausmacht. Nach der Erzeugung

Tabelle 3-1: Beschreibung der Klasse *projTree*

Klasse <i>projTree</i>		
Attribut	Typ	Beschreibung
Root	projVector	Initialer Knoten
NoParams	integer	Anzahl Parameter
ParamMatrices	cell	Matrizen für die Rekursion
MatrixAnnot	cell	Beschriftungen für grafische Ausgabe
MaxDepth	integer	Maximale Tiefe des Baumes
L, U, P, Pw	double	Matrizen aus LU-Zerlegung
V	double	Projektionsmatrix
ε_D	double	Abbruchschranke für Deflation
Methoden		Beschreibung
projTree(Root,NoParams)		Konstruktor
AddNodeToTree		Äußere Funktion zum Hinzufügen eines neuen Blattes, interner Aufruf von AddNode
AddNode		Hinzufügen eines neuen Blattes
orthNode		Orthogonalisierung der Projektionsvektoren
plotTree		Grafische Ausgabe der Projektionsvektoren

und Initialisierung von zwei Instanzen der Klasse *projTree* wird lediglich eine von der gewünschten Anzahl der Basisvektoren q abhängende for-Schleife ausgeführt. Innerhalb dieser ruft jede Instanz von *projTree* ihre Methode *addNodeToTree* auf, um einen weiteren Projektionsvektor zur Basis hinzuzufügen. Die Projektionsmatrizen selbst stellen ein öffentliches, von außen zugängliches Attribut der Klasse dar und können daher am Ende direkt ausgelesen werden.

Der Aufbau und die Methoden der beiden Klassen orientieren sich eng an der rekursiven Abhängigkeit der Momente. Das Grundprinzip beruht auf der Beobachtung, dass jeder neue Basisvektor nur von seinem direkten Vorgänger innerhalb der Baumstruktur abhängt und über eine Matrix-Vektor Multiplikation mit den Koeffizientenmatrizen M_i aus (3-3) berechnet wird.

Dies wird durch die Klasse *projVector* (Tabelle 3-2) realisiert. In jeder Instanz werden neben dem Basisvektor selbst, abgelegt im Attribut *VectData*, die Matrizen M_i als „cell-array“¹ in *ParamMatrices* gespeichert. Die unterlagerten, neu zu erzeugenden Projektionsvektoren sind ebenfalls Elemente der Klasse *projVector*. Auf sie kann zugegriffen werden, indem ihre Adressierungen als „Zeiger“ im Attri-

¹Ein „cell-array“ ist ein spezieller Matlab Datentyp. Die Elemente eines „cell-array“ können aus beliebigen anderen Datentypen, z. B. Matrizen, Text oder Datenstrukturen, bestehen.

Tabelle 3-2: Beschreibung der Klasse *projVector*

Klasse <i>projVector</i>		
Attribut	Typ	Beschreibung
Vectdata	double	Numerische Daten des Projektionsvektors
Children	handle	Zeiger auf zugehörige Blätter
MaxChildren	integer	Maximale Anzahl zugehöriger Blätter
NoChildren	integer	Momentane Anzahl zugehöriger Blätter
ParamMatrices	cell	Matrizen zur Erzeugung der Blätter
MatrixAnnot	cell	Beschriftung für grafische Ausgabe
Annotation	char	Eigene Beschriftung
Methoden		Beschreibung
<i>projVector</i> (MaxChildren, VectData)		Konstruktor
addChild		Blatt hinzufügen
deleteChild		Blatt löschen
destroyChild		Blatt-Typ entfernen

but *Children* hinterlegt werden. Auf diese Weise entsteht die gewünschte Baumstruktur, siehe Bild 3-2.

Neben dem Konstruktor enthält die Klasse *projVector* drei weitere, einfache Methoden. Mit *addChild* wird ein neues Blatt erzeugt, mit *deleteChild* wird ein vorhandenes Blatt und mit *destroyChild* ein Blatttyp entfernt. Die letztgenannte Methode wird zur Realisierung der Deflation benötigt. Ist eine unterlagerte Instanz, z. B. der Vektor M_2v in der Notation von Bild 3-2, linear abhängig von den übrigen Basisvektoren, so wird der zugehörige Knoten und auch die korrespondierende Matrix, hier im Beispiel die Matrix M_2 , im überlagerten Knoten entfernt. Auf diese Weise ist gewährleistet, dass im Rahmen der Deflation stets der gesamte Teilbaum, d. h. auch alle unterlagerten Knoten, entfernt wird.

Die Entscheidung, ob ein Projektionsvektor linear abhängig ist, kann jedoch nicht allein auf Basis der Klasse *projVector* getroffen werden, da hierzu die gesamte Baumstruktur berücksichtigt werden muss. Aus diesem Grund wurde die zweite Klasse *projTree* definiert. Sie enthält als Attribute den Zeiger auf das oberste Element, die Wurzel, des *projVector*-Baumes und ebenfalls die allgemeinen Systeminformationen über die Anzahl der Parameter und die zugehörigen Systemmatrizen. Zur Erhöhung der numerischen Robustheit werden statt einer Multiplikation mit $\tilde{A}_0^{-1}(s_0) := (A_0 - s_0 I)^{-1}$ entsprechende Gleichungssysteme gelöst. Hierfür wird eine LU-Zerlegung (siehe z. B. [Wer92])

$$\tilde{A}_0(s_0) = P \cdot L \cdot U$$

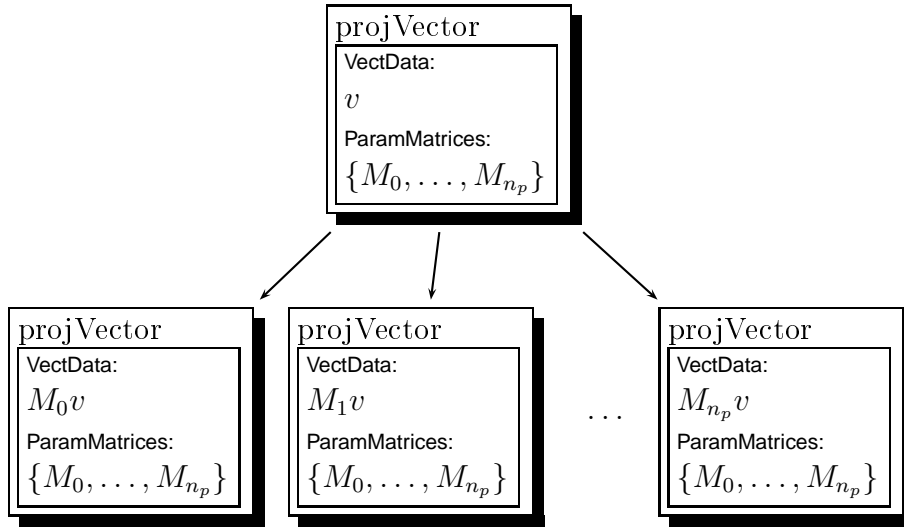


Bild 3-2: Verknüpfung von Projektionsvektoren innerhalb des objektorientierten Arnoldi-Algorithmus

der Matrix $\tilde{A}_0(s_0)$ verwendet. Alle innerhalb der Baumstruktur erzeugten Projektionsvektoren werden orthogonalisiert und in der Variablen V abgelegt.

Für die Erstellung der Baumstruktur ist es wichtig, dass zuerst alle Blätter einer bestimmten Rekursionstiefe erzeugt werden, bevor das erste Element der nächsthöheren Ebene erstellt wird. Dies entspricht einer sukzessiven Erhöhung der Ordnung der übereinstimmenden Momente bei der MOR. Die Realisierung erfolgt über die Variable *MaxDepth* und das Zusammenspiel der beiden Methoden *AddNode* und *AddNodeToTree*. Wie bereits dargestellt, wird von Algorithmus 3 die Methode *AddNodeToTree* aufgerufen. Diese führt ihrerseits *AddNode* aus, wodurch der nächste Knoten erstellt wird. Liefert *AddNode* keinen neuen Knoten zurück, wird die maximale Rekursionstiefe erhöht und erneut *AddNode* gestartet.

Das Hinzufügen eines Knotens basiert auf einer einfachen Fallunterscheidung. Ist die maximale Anzahl Blätter eines Knotens noch nicht erreicht, wird ein neues Blatt erzeugt. Andernfalls wird an den vorhandenen Blättern versucht, ein Blatt zu erzeugen. Dabei gilt die Einschränkung, dass die maximale Rekursionstiefe noch nicht erreicht sein darf.

Nachdem ein neuer Knoten erzeugt wurde, wird der Projektionsvektor mittels der Methode *OrthNode* orthogonalisiert und mit Hilfe der Deflationsschranke ε wird geprüft, ob er als neuer Basisvektor in die Projektionsmatrix V übernommen wird. Liegt eine lineare Abhängigkeit vor, wird der zugehörige Knoten entfernt und der nächste Knoten erzeugt.

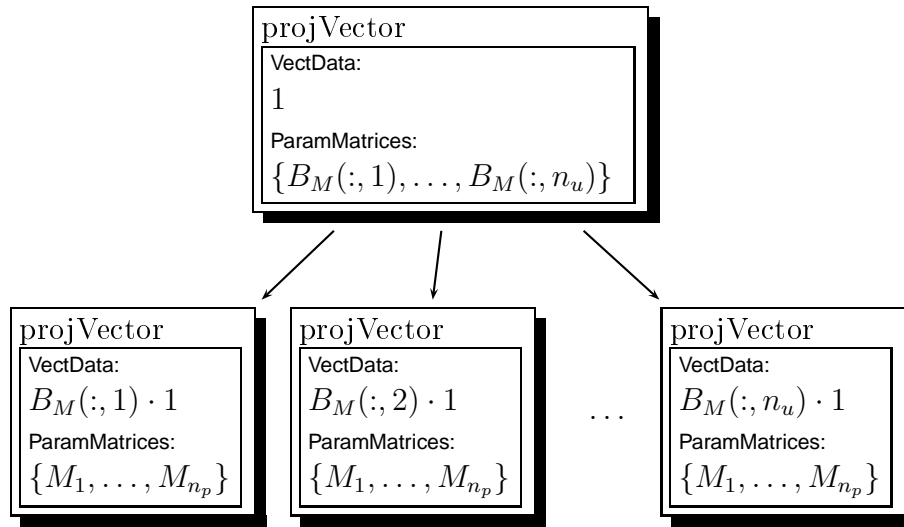


Bild 3-3: Wurzel der eingangsseitigen Baumstruktur

Die Methode *AddNode* ist vollständig rekursiv implementiert, weshalb sie ohne globale Überwachungs- und Steuerungsindizes auskommt. Im Vergleich zu einer Realisierung mit ineinander verschachtelten for-Schleifen ist diese Implementierung wesentlich überschaubarer.

Damit im Rahmen der MOR beliebige Ordnungen des reduzierten Systems möglich sind, dürfen nur Vektoren als *VectData* in der Klasse *projVector* verwendet werden. Bei mehreren Ein- und Ausgängen bestehen die Momente jedoch aus Matrizen, siehe erneut Bild 3-1. Dies hat zur Folge, dass für jede Spalte in B_M und jede Zeile in C_M ein eigener „vektorieller“ Rekursionsbaum erzeugt werden muss. Mit Hilfe der hier vorgestellten dynamischen Datenstrukturen können diese jedoch zu einer einzigen Baumstruktur zusammengefasst werden. Dazu muss lediglich der initiale Knoten, die Wurzel des Baumes, gesondert parametrisiert werden. Im Gegensatz zu dem allgemeinen Aufbau, der in Bild 3-2 dargestellt ist, werden für den initialen Knoten statt der Koeffizientenmatrizen M_i die Spalten von B_M bzw. die Zeilen von C_M verwendet. Der Unterschied ist in Bild 3-3 für B_M verdeutlicht. Indem man die Variable *VectData* auf 1 setzt, bleiben sämtliche Methoden und Bestandteile der Rekursion, sowohl von *projVector*, als auch von *projTree*, unberührt. Diese einfache Behandlung von parametrischen Mehrgrößensystemen stellt einen weiteren Vorteil des objektorientierten Arnoldi-Algorithmus dar.

Eine grafische Ausgabe der Baumstruktur wurde durch die Methode *plotTree* implementiert. Sie ist insbesondere bei der Analyse des Reduktionsergebnisses hilfreich, da auf einfache Weise geprüft werden kann, welche Teile der Momente

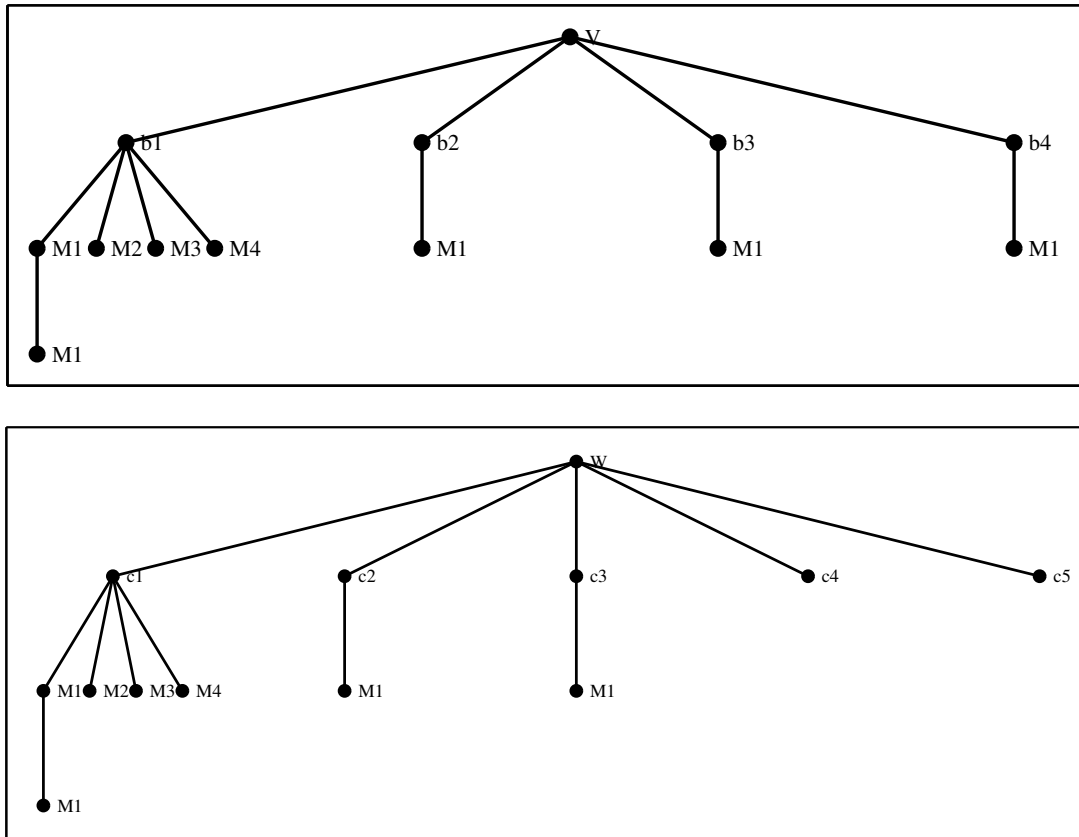


Bild 3-4: Grafische Ausgabe der verwendeten Basisvektoren innerhalb des objekt-orientierten Arnoldi-Algorithmus

als Basisvektoren verwendet wurden und welche Vektoren linear abhängig waren. Hieraus wird auch ersichtlich, bis zu welcher Ordnung die Momente abgeglichen werden. Eine beispielhafte Ausgabe ist in Bild 3-4 zu finden. Sie beruht erneut auf einer Reduktion des Aktormoduls, das bereits in Abschnitt 2.5 als Anwendungsbeispiel genutzt wurde. In diesem Fall liegt jedoch ein von drei Parametern abhängiges System vor.

Zur Illustration der grafischen Ausgabe wurde eine Reduktion mit einem Entwicklungspunkt $s_0 = 0$ auf die Ordnung 12 durchgeführt. Oben in Bild 3-4 befindet sich die Baumstruktur für den eingangsseitigen Krylov-Unterraum (Projektionsmatrix V), unten für den ausgangsseitigen (Projektionsmatrix W). Zunächst ist anhand der Darstellung erkennbar, dass für beide Projektionsmatrizen die Rekursionstiefe drei erreicht wird bzw. erforderlich ist. Da das betrachtete System über sechs Eingänge und fünf Ausgänge verfügt, ist dies bemerkenswert, denn bei drei Parametern stehen insgesamt 30 (eingangsseitig) bzw. 25 (ausgangsseitig) potentielle Projektionsvektoren bis zur Rekursionstiefe 2 zur Verfügung. Daraus lässt

sich zum einen erkennen, dass eine Reihe von Projektionsvektoren linear abhängig sind. Im eingangsseitigen Fall werden bereits auf der ersten Ebene von den sechs Spalten von B_M nur vier verwendet. Beim ausgangsseitigen Unterraum werden jedoch alle zu den fünf Ausgängen gehörenden Spalten verwendet. Zum anderen folgt aus der linearen Abhängigkeit auch, dass mit dem vorliegenden reduzierten System alle Momente – sowohl die zur Laplace-Variablen s , als auch zu den zusätzlichen Parametern gehörenden – mindestens bis zur Ordnung drei abgeglichen werden. Dies resultiert daraus, dass beide Projektionsmatrizen einzeln betrachtet die Momente der Ordnung 0 und 1 abgleichen und sich bei einem zweiseitigen Reduktionsverfahren die abgeglichenen Momente addieren.

Auffällig ist für beide Projektionsmatrizen, dass die zu den Parametern gehörenden Koeffizientenmatrizen M_2 , M_3 , M_4 nur jeweils einmal verwendet werden. Die Übrigen mit ihnen erzeugten Basisvektoren sind linear abhängig zu der vorhandenen Basis. Dieser Effekt konnte bei den meisten im Rahmen dieser Arbeit betrachteten Systemen beobachtet werden. Er ist wahrscheinlich darauf zurückzuführen, dass die Parameter in den hier betrachteten Anwendungen nur auf wenige Komponenten der gesamten Systemmatrix einen Einfluss besitzen. Daher sind in den Matrizen A_i ($i \geq 1$) nur wenige Einträge ungleich Null. Bei der Matrix-Vektor Multiplikation im Rahmen der Erstellung der Baumstruktur führt dies anscheinend oftmals zur linearen Abhängigkeit.

Der hier vorgestellte objektorientierte Arnoldi-Algorithmus stellt ein numerisch robustes Verfahren zur Berechnung von Projektionsmatrizen für die Modellordnungsreduktion parametrischer Systeme dar und erfüllt die eingangs erwähnten Anforderungen. Der Entwicklungspunkt s_0 , von dem die Krylov-Unterräume (3-4) und (3-6) abhängen, kann frei vorgegeben werden. Es können beliebige Ordnungen des reduzierten Systems realisiert werden, da die Basisvektoren einzeln zu den Projektionsmatrizen hinzugefügt werden. Auch die Deflation wird durch den rekursiven Aufbau und die Verwendung einer Baumstruktur für die Projektionsmatrizen realisiert. Mehrere Entwicklungspunkte können durch eine mehrfache Ausführung des objektorientierten Arnoldi-Algorithmus verwendet werden. Die separat berechneten Projektionsmatrizen werden - analog zum nichtparametrischen Fall - abschließend durch eine Orthonormalisierung zusammengefasst. Ein Nachteil des hier präsentierten Ansatzes besteht darin, dass sämtliche Parameter des Algorithmus wie die Ordnung des reduzierten Systems oder die Entwicklungspunkte manuell vorgegeben werden müssen. In Abhängigkeit von dem zu reduzierenden System, kann dies erheblichen Aufwand verursachen.

3.2 Matrix Interpolation

Die Matrix Interpolation stellt den zweiten innerhalb dieser Arbeit verwendeten Ansatz zur MOR parametrischer Systeme dar. Im Gegensatz zu der im vorher-

gehenden Abschnitt vorgestellten manuellen rationalen Interpolation parametrischer Systeme handelt es sich bei der Matrix Interpolation nicht um eine Weiterentwicklung eines bestimmten nichtparametrischen Reduktionsverfahrens. Die Matrix Interpolation basiert zwar auf Reduktionsverfahren für nichtparametrische Systeme, allerdings können alle auf Projektionsmatrizen basierenden Verfahren eingesetzt werden.

Entstanden ist die Matrix Interpolation durch die Übertragung eines Reduktionsansatzes für nichtlineare Systeme, der sogenannten TPWL (Trajectory Piecewise Linear), auf parametrische Systeme. Die TPWL, erstmals beschrieben in [RW03], verfolgt den Ansatz, eine festgelegte Trajektorie, die Trainings-Trajektorie, zu definieren und entlang dieser Trajektorie an vordefinierten Punkten im Zustandsraum lineare Systeme zu berechnen. Diese werden nachfolgend separat mit den Methoden der MOR für lineare Systeme reduziert, siehe [VRW03] für den beispielhaften Einsatz des balancierten Abschneidens und [BSMM08] für eine Anwendung der Proper Orthogonal Decomposition (POD). Zur beschleunigten Simulation des nichtlinearen Systems werden die reduzierten Systeme zeit- oder zustandsabhängig mit Hilfe von Gewichtsfunktionen überlagert. Dieser Ansatz ist weiterhin Gegenstand der Forschung. Beispielsweise wird versucht, die Approximationsgüte durch geeignete Gewichtsfunktionen und gut platzierte Stützstellen zu verbessern, siehe z. B. [AF10].

In [LK05] sowie [LLL⁺05] wird diese Idee auf parametrische Systeme übertragen. Statt Punkte im Zustandsraum auszuwählen, werden Parameterstützstellen festgelegt, an denen reduzierte Systeme berechnet werden. Um ein parametrisches reduziertes System zu erhalten, werden die einzelnen Projektionsmatrizen mit Hilfe einer Singulärwertzerlegung zu einem gemeinsamen Paar von Projektionsmatrizen zusammengefasst. Tendenziell führt dies zu einem parametrischen reduzierten System mit geringerer Systemordnung, verglichen mit der manuellen rationalen Interpolation parametrischer Systeme, da die Momente bzgl. der Parameter für den Reduktionsprozess keine Rolle spielen. Genau wie die manuelle rationale Interpolation erfordert dieser Ansatz eine affin-lineare oder zumindest polynomielle Parameterabhängigkeit. Systeme mit anderen Parameterabhängigkeiten können unter Umständen nicht reduziert werden. Auch diese Herangehensweise wird weiterhin verfolgt, so ist beispielsweise in [BBBG11] dieser Ansatz in Verbindung mit der \mathcal{H}_2 -optimalen Interpolation zu finden.

Die in dieser Arbeit eingesetzte Variante der Matrix Interpolation ist noch enger mit der TPWL verbunden. Statt eine gemeinsame Projektionsmatrix für das parametrische reduzierte System zu berechnen, werden an den Stützstellen separat reduzierte Systeme aufgestellt. Diese werden dann nachfolgend überlagert, d. h. interpoliert, um ein parametrisches reduziertes System zu erhalten. Dieser Ansatzes ist in [PMEL10] zu finden. Die folgenden Ausführungen orientieren

sich ebenfalls an diesem Artikel. Eine Art Vorläufer dieser Variante der Matrix Interpolation bildet der Artikel [BB09], in dem die Übertragungsfunktionen zu verschiedenen Parameterwerten mit Hilfe von Lagrange-Polynomen interpoliert werden. Aussagen über die Systeme im Zeitbereich werden jedoch nicht getroffen. Der Schwerpunkt liegt stattdessen auf der Verwendung dünner Gitter zur Diskretisierung des Parameterraums.

Neuere Arbeiten greifen die hier verwendete Form der Matrix Interpolation auf, beispielsweise im Kontext mechanischer Systeme in [GPS11]. Es ist zudem eine stabilitätserhaltende Methode entstanden, die allerdings auf einseitigen Projektionen basiert und somit die einsetzbaren Verfahren der MOR stark einschränkt, siehe [ECSP⁺11]. Da die Stabilität der reduzierten Systeme im Rahmen dieser Arbeit kein Problem darstellte, wird diese Modifikation nicht näher betrachtet, insbesondere weil hiermit auch der Einsatz der \mathcal{H}_2 -optimalen Interpolation im Rahmen der PMOR nicht mehr möglich ist.

Als Ausgangspunkt für die im Folgenden beschriebene Matrix Interpolation dient das wesentlich allgemeinere parametrische lineare System

$$\begin{aligned} E\dot{x} &= A(p)x + B(p)u, \\ y &= C(p)x, \end{aligned}$$

mit dem Parametervektor $p \in \mathbb{R}^{n_p}$. Die Art der Parameterabhängigkeit muss nicht näher bekannt sein, was insbesondere bei der Verwendung von CAE-Tools zur Erstellung des Modells von Vorteil ist. Stattdessen müssen, wie bereits erwähnt, Parameterstützstellen $p_i \in \mathbb{R}^{n_p}$, $1 \leq i \leq n_S$ definiert werden, mit den zugehörigen Systemen

$$\begin{aligned} E_i\dot{x} &= A_i x + B_i u, \\ y &= C_i x. \end{aligned} \tag{3-7}$$

Diese Systeme werden zunächst unabhängig voneinander reduziert, und zwar auf die selbe Ordnung q . Wie in Abschnitt 2.2 beschrieben, geschieht dies durch die Projektion mit den beiden Projektionsmatrizen $V_i \in \mathbb{R}^{n,q}$ und $W_i \in \mathbb{R}^{n,q}$. Das eingesetzte Verfahren zur Berechnung von V_i und W_i spielt hierbei keine Rolle. Es sind sowohl die manuelle rationale Interpolation aus Abschnitt 2.3, die \mathcal{H}_2 -optimale Interpolation und prinzipiell auch weitere MOR Verfahren einsetzbar. Das Reduktionsverfahren kann zudem an allen Diskretisierungspunkten unabhängig voneinander gewählt werden. Analog zu Gleichung (2-8) erhält man die auf Ordnung q reduzierten Systeme

$$\begin{aligned} \underbrace{W_i^T E_i V_i}_{E_{r,i}} \dot{x}_r(t) &= \underbrace{W_i^T A_i V_i}_{A_{r,i}} x_r(t) + \underbrace{W_i^T B_i}_{B_{r,i}} u(t), \\ y_r(t) &= \underbrace{C_i V_i}_{C_{r,i}} x_r(t). \end{aligned} \tag{3-8}$$

Die Parameterabhängigkeit wird nun im reduzierten Fall über eine Interpolation der reduzierten Systemmatrizen hergestellt. Hierzu sind Gewichtsfunktionen

$$\omega_i : \mathbb{R}^{n_p} \rightarrow [0, 1], \quad p \mapsto \omega_i(p), \quad 1 \leq i \leq n_S$$

zu definieren mit $\sum_{i=1}^{n_S} \omega_i(p) = 1$ und $\omega_i(p_i) = 1$. Die parametrischen reduzierten Systemmatrizen ergeben sich dann als

$$\begin{aligned} E_r(p) &= \sum_{i=1}^{n_S} \omega_i(p) E_{r,i}, & A_r(p) &= \sum_{i=1}^{n_S} \omega_i(p) A_{r,i}, \\ B_r(p) &= \sum_{i=1}^{n_S} \omega_i(p) B_{r,i}, & C_r(p) &= \sum_{i=1}^{n_S} \omega_i(p) C_{r,i}. \end{aligned} \tag{3-9}$$

Die Beschaffenheit der Funktionen ω_i stellt einen Freiheitsgrad des Reduktionsverfahrens dar und kann nicht allgemeingültig angegeben werden. Für diese Arbeit ergeben sich die Gewichtsfunktionen als Nebenprodukt der in Kapitel 5 vorgestellten Interpolation von Paretomengen, basierend auf einer Spline-Interpolation. Daher wird auf die Wahl von ω_i an dieser Stelle nicht näher eingegangen.

Die Interpolation (3-9) von Systemmatrizen wird im Allgemeinen nicht zu zufriedenstellenden Ergebnissen führen. Bereits durch eine Vertauschung der Reihenfolge der Zustandsgleichungen für verschiedene Parameterstützstellen entstehen außerhalb der Stützstellen Systeme mit vollkommen anderem Systemverhalten, auch wenn die einzelnen Systeme das Ausgangssystem sehr gut approximieren. Weitere Ausführungen hierzu sowie ein einfaches, anschauliches Beispiel mit zwei Zuständen sind in [PMEL10] zu finden.

Für die Ausgangssysteme (3-7) ließe sich voraussetzen, dass die Zustandsvariablen die gleiche physikalische Bedeutung besitzen und identisch angeordnet sind. Aber selbst in diesem Fall lässt sich über die Anordnung und Bedeutung der Zustände nach der Reduktion keine Aussage mehr treffen.

Abhilfe schafft hier die im Folgenden vorgestellte Methode, mit der die unterschiedlichen reduzierten Systeme „kompatibel“ zueinander gemacht werden, in dem Sinn, dass die Interpolation der reduzierten Systeme (3-9) möglich wird. Dafür werden für jedes reduzierte System zwei reguläre Matrizen $T_i \in \mathbb{R}^{q,q}$ und $M_i \in \mathbb{R}^{q,q}$ berechnet und die reduzierten Systeme (3-8) modifiziert zu

$$\begin{aligned} M_i E_{r,i} T_i^{-1} \dot{x}_{r,i}^* &= M_i A_{r,i} T_i^{-1} x_{r,i}^* + M_i B_{r,i} u, \\ y &= C_{r,i} T_i^{-1} x_{r,i}^*. \end{aligned} \tag{3-10}$$

Dies entspricht einer Zustandstransformation $x_{r,i}^* = T_i x_{r,i}$ und einer nachfolgenden Multiplikation der Zustandsgleichung von links mit M_i . Festzuhalten ist an

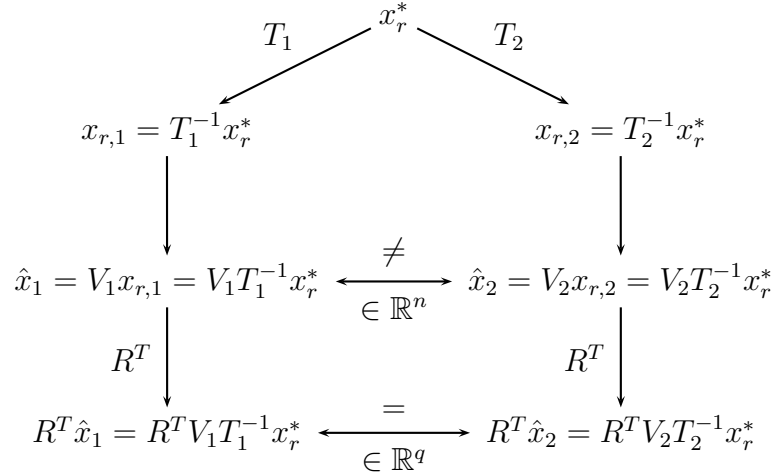


Bild 3-5: Übersicht über Zustandstransformation, Projektion und Rückprojektion, vgl. [PMEL10]

dieser Stelle, dass das Übertragungsverhalten und die dynamischen Eigenschaften der reduzierten Systeme unverändert bleiben.

Das mit Hilfe der Matrizen T_i zu erreichende Ziel besteht darin, den Zustandsvektoren eine gemeinsame Bedeutung zu geben. Mathematisch bedeutet dies, dass Zustandstransformationen T_i gesucht sind, die die Zustandsvektoren der reduzierten Systeme bzgl. einer gemeinsamen Basis

$$x_{r,1}^* = x_{r,2}^* = \dots = x_{r,n_S}^* =: x_r^*$$

ausdrücken. Für die Interpolation der Systemmatrizen ist entscheidend, dass die Zustandsvektoren im ursprünglichen Zustandsraum \mathbb{R}^n eine gemeinsame Bedeutung haben. Da jedoch für jedes reduzierte System die Projektion

$$\hat{x}_i = V_i x_{r,i} \in \mathbb{R}^n$$

verwendet wird, befinden sich die Zustände im jeweiligen Unterraum, der von den Spalten von V_i aufgespannt wird, unabhängig von der gewählten Transformation T_i . Dies hat zur Folge, dass eine gemeinsame Bedeutung im \mathbb{R}^n nicht hergestellt werden kann, da die Vektoren $V_i x_r^*$ stets in verschiedenen Unterräumen liegen. Es ist allerdings möglich, die Zustandsvektoren bzgl. eines weiteren Unterraums, aufgespannt von den Spalten einer Matrix $R \in \mathbb{R}^{n,q}$ vergleichbar zu machen, indem eine Rückprojektion mit R^T vorgenommen wird. Bild 3-5 veranschaulicht diesen Zusammenhang. Ausgehend von einem Zustandsvektor x_r^* wird die Transforma-

tion in das lokale Koordinatensystem $x_{r,i} = T_i^{-1}x_r^*$ durchgeführt. Das Ergebnis wird mittels V_i in den Zustandsraum des Ausgangssystems überführt:

$$\hat{x}_i = V_i x_{r,i} = V_i T_i^{-1} x_r^*.$$

Die resultierenden Vektoren werden im Allgemeinen voneinander verschieden sein. Multipliziert man jetzt jedoch \hat{x}_i mit R^T , woraus sich

$$R^T \hat{x}_i = R^T V_i x_{r,i} = R^T V_i T_i^{-1} x_r^*$$

ergibt, so sind diese Vektoren identisch im \mathbb{R}^q , falls als Transformation

$$T_i = R^T V_i \tag{3-11}$$

verwendet wird. Die Zustandsräume der reduzierten Systeme $x_{r,i}^*$ heißen dann kompatibel zueinander bzgl. der Matrix $R \in \mathbb{R}^{n,q}$, siehe [PMEL10]. Die gemeinsamen Zustände sind entsprechend

$$x_r^* = T_i x_{r,i} = R^T V_i x_{r,i}, \quad (1 \leq i \leq n_S).$$

Zur Festlegung der Matrix R können die Projektionsmatrizen V_i verwendet werden. Sie repräsentieren jeweils die für die Systemdynamik relevantesten Unterräume an den zugehörigen Parameterstützstellen p_i . Es ist nun von Vorteil, wenn als Unterraum, bzgl. dessen die reduzierten Systeme kompatibel sind, die für alle Stützstellen insgesamt relevantesten Richtungen zu verwenden. Dies lässt sich über eine Singulärwertzerlegung der Matrix

$$V_{all} = [V_1, V_2, \dots, V_{n_S}],$$

die von allen Projektionsmatrizen gebildet wird, erreichen. Die q Spalten der Matrix U_{SVD} , die zu den größten Singulärwerten der Singulärwertzerlegung

$$V_{all} = U_{SVD} \Sigma_{SVD} V_{SVD}$$

gehören, werden als Matrix R genutzt.

Die zweite Matrix M_i in (3-10) wird gemäß [PMEL10] ebenfalls mit Hilfe der Matrix R berechnet als

$$M_i = (W_i^T R)^{-1}. \tag{3-12}$$

Die Begründung für diese Wahl benötigt einige Vorbetrachtungen, die eine andere Sichtweise auf die MOR darstellen, als sie bisher verwendet wurde.

Wie bereits mehrfach erwähnt besteht der erste Schritt der Reduktion darin, den hochdimensionalen Zustandsvektor x durch die Projektion $x = Vx_r$ zu ersetzen. Eingesetzt in die Zustandsgleichung des Ausgangssystems ergibt sich

$$EV\dot{x}_r(t) = AVx_r(t) + Bu(t) + \varepsilon(t)$$

mit einem Residuum $\varepsilon(t)$, da es sich um ein überbestimmtes Gleichungssystem handelt, das im Allgemeinen nicht lösbar ist. Durch die anschließende Multiplikation mit der zweiten Projektionsmatrix W^T erhält man das eindeutig lösbare System

$$W^T EV\dot{x}_r(t) = W^T AVx_r(t) + W^T Bu(t). \quad (3-13)$$

Dies bedeutet für das Residuum, dass $W^T \varepsilon(t) = 0$ gelten muss. Das Residuum befindet sich demnach immer im orthogonalen Komplement W^\perp von W . Dies veranschaulicht auch den Einfluss von W auf die Approximationsgüte des reduzierten Systems.

Fügt man nun in Gleichung (3-13) die Identität $W^T S(W^T S)^{-1} = I$ ein, mit einer ebenfalls rechteckigen Matrix $S \in \mathbb{R}^{n,q}$, so erhält man

$$W^T \underbrace{S(W^T S)^{-1} W^T}_{\mathcal{P}_S} EV\dot{x}_r(t) = W^T \underbrace{S(W^T S)^{-1} W^T}_{\mathcal{P}_S} AVx_r(t) + W^T \underbrace{S(W^T S)^{-1} W^T}_{\mathcal{P}_S} Bu(t). \quad (3-14)$$

Die Matrix S kann beliebig gewählt werden, solange das Produkt $W^T S$ invertierbar ist. In der nun vorliegenden Gleichung taucht an drei Stellen eine besondere Art der Projektion in Form der Abbildung

$$\mathcal{P}_S : \mathbb{R}^n \rightarrow \mathbb{R}^n, x \mapsto S(W^T S)^{-1} W^T x = \mathcal{P}_S x$$

auf. Es handelt sich hierbei um die Projektion in den von den Spalten von S aufgespannten Unterraum entlang des orthogonalen Komplements W^\perp des von W aufgespannten Unterraums. Dies bedeutet zum einen, dass $\mathcal{P}_S x \in S$ sowie $(\mathcal{P}_S x - x) \in W^\perp$ gelten. Ersteres lässt sich sehr leicht einsehen, da

$$\mathcal{P}_S x = S(W^T S)^{-1} W^T x = Sy = \sum_{i=1}^q y_i s_i,$$

also $\mathcal{P}_S x$ als Linearkombination der Spalten s_i von S darstellbar ist. Für die zweite Aussage ist nachzuweisen, dass jeder Vektor aus W orthogonal zu $\mathcal{P}_S x - x$

ist. Dies ist erfüllt, wenn sämtliche Basisvektoren w_i , d. h. die Spalten von W , orthogonal zu $\mathcal{P}_S x - x$ sind und folgt aus

$$w_i^T (\mathcal{P}_S x - x) = w_i^T \mathcal{P}_S x - w_i^T x = \underbrace{w_i^T S (W^T S)^{-1} W^T x}_{e_i^T} - w_i^T x = w_i^T x - w_i^T x = 0.$$

Für das reduzierte System (3-14) ergibt sich nun die folgende Interpretation. Der Zustandsvektor des reduzierten Systems sowie dessen Ableitung werden mittels V in den Zustandsraum des Ausgangssystems projiziert und anschließend mit E bzw. A abgebildet. Das Ergebnis dieser Abbildungen sowie der Systemeingang Bu werden nun orthogonal zu W in den Unterraum S projiziert. Abschließend sorgt die erneute Multiplikation mit W^T in (3-14) für die Eindeutigkeit des reduzierten Systems. Wird nun in diesem letzten Schritt statt W^T die Matrix S^T verwendet und setzt man zusätzlich voraus, dass die Spalten von S orthogonal zueinander sind (d. h. $S^T S = I$), so entsteht das reduzierte System

$$(W^T S)^{-1} W^T E V \dot{x}_r = (W^T S)^{-1} W^T A V x_r + (W^T S)^{-1} W^T B u$$

Hiermit liegt nun die gewünschte Form des reduzierten Systems aus (3-10) vor, in der die Zustandsgleichung des reduzierten Systems von links mit einer quadratischen Matrix, hier $(W^T S)^{-1}$, multipliziert wird. Die Bedeutung dieser Multiplikation stellt, wie bereits dargelegt, eine Projektion der Zustandsgleichungen in den Unterraum S dar. Dies lässt sich separat für jedes reduzierte System (3-8) durchführen. Anschließend werden alle reduzierten Systeme in den gleichen Unterraum S projiziert. Da in der Matrix R die relevantesten Richtungen der Systemdynamik des Ausgangssystems enthalten sind und R zudem aus zueinander orthonormalen Spalten besteht, ist naheliegend $S = R$ zu wählen, womit sich die bereits angegebene Matrix M_i ergibt.

Zusammengefasst erhält man als Endergebnis das parametrische reduzierte System

$$\begin{aligned} E_r(p) \dot{x}_r &= A_r(p) x_r + B_r(p) u, \\ y &= C_r(p) x \end{aligned}$$

mit

$$\begin{aligned} E_r(p) &= \sum_{i=1}^{n_S} \omega_i(p) M_i E_{r,i} T_i^{-1}, & A_r(p) &= \sum_{i=1}^{n_S} \omega_i(p) M_i A_{r,i} T_i^{-1}, \\ B_r(p) &= \sum_{i=1}^{n_S} \omega_i(p) M_i B_{r,i}, & C_r(p) &= \sum_{i=1}^{n_S} \omega_i(p) C_{r,i} T_i^{-1}, \end{aligned} \tag{3-15}$$

wobei $E_{r,i}$, $A_{r,i}$, $B_{r,i}$, $C_{r,i}$ gemäß (3-8) durch separate MOR berechnet werden und T_i und M_i nach (3-11) bzw. (3-12) gewählt werden.

Die Approximationsgüte dieses parametrischen reduzierten Systems lässt sich nur schwer allgemein beschreiben. An den Parameterstützstellen selbst entspricht die Approximationsgüte der Güte der einzelnen reduzierten Systeme. Von Vorteil ist hier insbesondere im Vergleich zu der parametrischen manuellen rationalen Interpolation, dass die Anzahl der Parameter keinen Einfluss auf die Ordnung der reduzierten Systeme besitzt, was tendenziell zu parametrischen reduzierten Systemen mit geringerer Systemordnung führt.

Außerhalb der Stützstellen lässt sich die Approximationsgüte nicht vorhersagen. Sie hängt von der Lage der Stützstellen sowie von den Funktionen ω_i ab. Es ist lediglich zu erwarten, dass aus Stetigkeitsgründen in der Nähe einer Parameterstützstelle eine höhere Güte vorliegt, als bei größerem Abstand. Dies bestätigt sich auch weitgehend bei den im Rahmen dieser Arbeit durchgeführten Rechnungen.

In Tabelle 3-3 sind die beiden innerhalb dieser Arbeit verwendeten Ansätze zur parametrischen Modellordnungsreduktion, die manuelle rationale Interpolation parametrischer Systeme sowie die Matrix Interpolation, noch einmal zusammengefasst. Die manuelle rationale Interpolation parametrischer Systeme besitzt die gleichen Eigenschaften und Parameter wie im nichtparametrischen Fall. Zusätzlich zu beachten ist, dass nur Systeme mit einer affin-linearen Parameterabhängigkeit reduziert werden können. Die Matrix Interpolation besitzt demgegenüber den Vorteil, wesentlich allgemeiner anwendbar zu sein. Zum einen können Systeme mit beliebigen Parameterabhängigkeiten reduziert werden. Zum anderen kann die Matrix Interpolation mit beliebigen Reduktionsverfahren für nichtparametrische Systeme kombiniert werden, solange diese auf einer Projektion mit Hilfe der Matrizen V und W beruhen. Da zudem die Ordnung des parametrischen reduzierten Systems, insbesondere bei einer Abhängigkeit von mehreren Parametern, geringer ist als bei der manuellen rationalen Interpolation, führt die Matrix Interpolation zu besseren Reduktionsergebnissen. Allerdings nur unter der Voraussetzung, dass die zusätzlichen Freiheitsgrade, die Anzahl und Lage der Parameterstützstellen sowie die Gewichtsfunktionen ω_i , in geeigneter Weise bestimmt werden können.

Die Matrix Interpolation wird im Rahmen dieser Arbeit ausschließlich mit dem IRKA eingesetzt. Dies ermöglicht eine weitgehend automatisierte Berechnung der reduzierten Systeme an den Parameterstützstellen. Die zusätzlichen Freiheitsgrade, d. h. die Anzahl und Lage der Parameterstützstellen sowie die Gewichtsfunktionen entstehen als Nebenprodukt der in Kapitel 5 beschriebenen Interpolation von Paretomengen. Das dort beschriebene Vorgehen führt zu parametrischen Systemen, die auch über eine affin-lineare Parameterabhängigkeit verfügen, sodass

auch die manuelle rationale Interpolation parametrischer Systeme eingesetzt werden kann. Konkrete Resultate für beide Ansätze sind in Abschnitt 5.4 zu finden.

Tabelle 3-3: Innerhalb dieser Arbeit verwendete Verfahren der PMOR

Rationale Interpolation der Übertragungsfunktion		
Parametrisches Originalsystem		
	<u>manuelle Interpolation</u>	<u>Matrix Interpolation</u>
Algorithmus:	objektorientierter Arnoldi-Algorithmus	beliebiger Algorithmus (z.B. IRKA) + anschließende Interpolation der reduzierten Systeme
Parameter:	<ul style="list-style-type: none"> • Deflationsschranke ε • Entwicklungspunkte s_i • Anzahl Basisvektoren q_i 	<ul style="list-style-type: none"> • Parameter des Reduktionsverfahrens • Anzahl n_S der reduzierten Systeme • zugehörige Parameterstützstellen p_i, $1 \leq i \leq n_S$ • Gewichtsfunktionen ω_i, $1 \leq i \leq n_S$
Eigenschaften:	<ul style="list-style-type: none"> ✓ Einfluss auf Approximationseigenschaften (z.B. stat. Genauigkeit) ✓ Ableitungen nach den Parametern können einfach berücksichtigt werden ✗ aufwändige manuelle Auswahl von s_i und q_i ✗ nur bei affin-linearer Parameterabhängigkeit anwendbar ✗ Ordnung des reduzierten Systems erhöht sich bei steigender Zahl von Parametern 	<ul style="list-style-type: none"> ✓ mit beliebigen nichtparametrischen Reduktionsverfahren kombinierbar ✓ Art der Parameterabhängigkeit ist beliebig ✓ geringere Ordnung des reduzierten Systems bei vielen Parametern ✗ Anzahl und Lage der Stützstellen muss manuell gewählt werden ✗ Bestimmung geeigneter Gewichtsfunktionen bei vielen Parametern aufwändig

4 Mehrzieloptimierung selbstoptimierender Systeme

In diesem Kapitel werden die Zusammenhänge vorgestellt, die für eine Verknüpfung der Mehrzieloptimierung mit der in den beiden vorhergehenden Kapiteln beschriebenen PMOR notwendig sind. Dazu gehört ein kurzer Überblick über die Grundlagen der Mehrzieloptimierung in Abschnitt 4.1 sowie eine Darstellung der hierarchischen Optimierung in Abschnitt 4.2 in der in dieser Arbeit verwendeten Form. Für die Verzahnung der hierarchischen Optimierung mit der PMOR sind die in der Optimierung verwendeten Modelle von entscheidender Bedeutung. Daher wird eine allgemeine Modellstruktur für die auf Simulationen basierte Mehrzieloptimierung selbstoptimierender Systeme in Abschnitt 4.3 dargestellt. Im letzten Abschnitt dieses Kapitels werden zwei Mehrzieloptimierungsprobleme und ihre Lösungsmengen aus dem Kontext des Feder-Neige-Prüfstands präsentiert. Die vorgestellten Ergebnisse bilden die Grundlage für die Anwendungsbeispiele in den weiteren Kapiteln.

4.1 Grundlagen der Mehrzieloptimierung

Die Mehrzieloptimierung ist seit langem Gegenstand intensiver Forschung. Erste Ansätze gehen bereits auf das 18. Jahrhundert zurück. Seitdem sind zahlreiche Anwendungsbeispiele, Beiträge zur Theorie und algorithmische Weiterentwicklungen entstanden. Eine Übersicht über den aktuellen Stand der Technik würde den Rahmen der vorliegenden Arbeit sprengen. Einen guten Überblick über etablierte Methoden sowie fast 700 Verweise auf weiterführende Literatur sind in [Mie99] zu finden. Die Darstellung innerhalb dieses Abschnitts beschränkt sich auf die Einführung der für das Verständnis dieser Arbeit notwendigen Zusammenhänge. Zudem werden die verwendeten Algorithmen kurz präsentiert.

Das Ziel der Mehrzieloptimierung besteht aus der gleichzeitigen Minimierung (oder Maximierung) mehrerer Zielfunktionen

$$f_i : \mathbb{R}^{n_p} \rightarrow \mathbb{R}, p \mapsto f_i(p), \quad 1 \leq i \leq n_o$$

die im Zielfunktionsvektor $F(p) = [f_1(p), \dots, f_{n_o}(p)]$, ($n_o \geq 2$) zusammengefasst werden. Für alle folgenden Betrachtungen wird angenommen, dass die Zielfunktionen zweimal stetig differenzierbar sind. Zum Vergleich von zwei Vektoren $u, v \in \mathbb{R}^{n_o}$ wird die folgende Ordnungsrelation eingeführt. Der Vektor u ist kleiner oder gleich v , d. h. $u \leq v$, falls

$$u_i \leq v_i, \quad 1 \leq i \leq n_o$$

gilt. Mit Bezug zu dieser Ordnungsrelation ist ein Mehrzieloptimierungsproblem (MOP) definiert als

$$\min_p \{F(p) : p \in \mathcal{D} \subseteq \mathbb{R}^{n_p}\}. \quad (4-1)$$

Der Definitionsbereich \mathcal{D} ist innerhalb dieser Arbeit ausschließlich durch obere und untere Schranken,

$$p_{i,\min} \leq p_i \leq p_{i,\max}, \quad 1 \leq i \leq n_p$$

festgelegt. Die Lösung des MOP (4-1) ist durch die sogenannten nichtdominierten bzw. paretooptimalen Punkte $p^* \in \mathcal{D}$ gegeben. Ein Vektor u dominiert hierbei einen Vektor v , falls

$$u \leq v \text{ und } u_i < v_i \text{ für mindestens ein } i \in \{1, \dots, n_p\}.$$

Alle nichtdominierten Punkte zusammen bilden die Paretomenge

$$P_F = \{p^* \in \mathcal{D} : \nexists p \in \mathcal{D} \text{ mit } F(p) \text{ dominiert } F(p^*)\}.$$

Gerade bei der praktischen Berechnung von Paretopunkten werden oftmals nur lokale Paretopunkte als Ergebnis zurückgegeben. Diese unterscheiden sich von den bisher definierten globalen Paretopunkten in der Hinsicht, dass nur innerhalb einer Umgebung $\mathcal{U} \subseteq \mathcal{D}$, mit $p^* \in \mathcal{U}$, der Paretopunkt p^* ein nichtdominierter Punkt ist. Jeder globale Paretopunkt ist natürlicherweise auch ein lokaler Paretopunkt. Die Umkehrung gilt, falls die Zielfunktionen f_i sowie der Definitionsbereich \mathcal{D} konvex sind. Eine Funktion f_i ist hierbei konvex, falls für zwei beliebige Vektoren $p^1, p^2 \in \mathbb{R}^{n_p}$ und $0 \leq \beta \leq 1$ die Ungleichung

$$f_i(\beta p^1 + (1 - \beta)p^2) \leq \beta f_i(p^1) + (1 - \beta)f_i(p^2)$$

erfüllt ist.

Jeder (lokale und globale) Paretopunkt erfüllt die sogenannte Karush-Kuhn-Tucker Bedingung (KKT-Bedingung), siehe [Kar39] und [KT51]. Diese stellt eine notwendige jedoch nicht hinreichende Bedingung der Paretooptimalität dar. Die KKT-Bedingung sagt aus, dass es für jeden paretooptimalen Punkt $p^* \in P_F$ eines MOP (4-1) ohne Nebenbedingungen einen Vektor $\gamma \in \mathbb{R}^{n_o}$ gibt mit

$$\gamma_i \geq 0, \quad (1 \leq i \leq n_o) \text{ und } \sum_{i=1}^{n_o} \gamma_i = 1,$$

sodass

$$\sum_{i=1}^{n_o} \gamma_i \nabla f_i(p^*) = 0$$

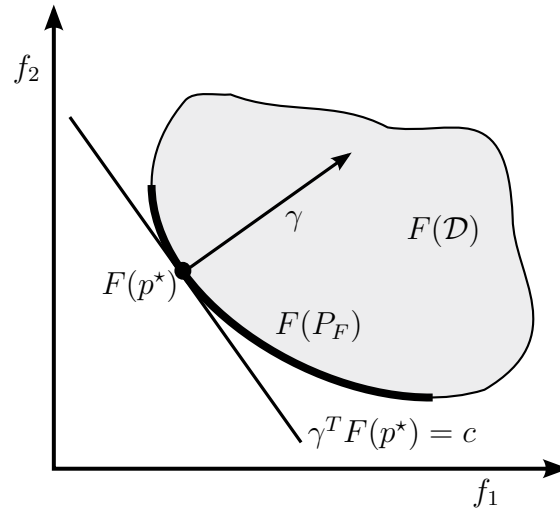


Bild 4-1: Paretofront für zwei Zielfunktionen und geometrische Bedeutung von γ

gilt. Für MOP mit Nebenbedingungen sind zusätzlich die Lagrange-Multiplikatoren und Ableitungen nach den aktiven Nebenbedingungen zu betrachten. Da im Rahmen dieser Arbeit jedoch nur MOP ohne zusätzliche Nebenbedingungen betrachtet werden, wird an dieser Stelle auf eine explizite Angabe der erweiterten KKT-Bedingung verzichtet. Weiterführende Details sind beispielsweise in [Hil01] zu finden. Die KKT-Bedingung besagt, dass die Gradienten der Zielfunktionen für jeden paretooptimalen Punkt p^* linear abhängig sind.

Mit Hilfe der KKT-Bedingung und Resultaten aus der Differentialgeometrie lässt sich zeigen, dass die paretooptimalen Punkte unter gewissen Voraussetzungen, z.B. $n_o \geq n_p$, eine $(n_o - 1)$ -dimensionale, differenzierbare Mannigfaltigkeit bilden. Der Beweis sowie die dafür notwendigen Grundlagen der Differentialgeometrie sind ebenfalls in [Hil01] nachzulesen. Eine wichtige Folgerung hieraus besteht darin, dass Paretopunkte nicht isoliert vorkommen können, sondern stets eine (zumindest lokal) zusammenhängende Lösungskurve bilden. Dies ist eine wesentliche Voraussetzung für die im Rahmen dieser Arbeit entwickelte parametrische Modellordnungsreduktion. Weiterhin ist beachtenswert, dass die Dimension $(n_o - 1)$ der Paretomenge nur von der Anzahl n_o der Zielfunktionen, aber nicht von der Anzahl n_p der Optimierungsparameter abhängt.

Die typische Form einer Paretomenge im Bildraum (für ein MOP mit zwei Zielfunktionen), d. h. das Resultat $F(P_F)$ der Anwendung von F auf die Paretomenge P_F , ist in Bild 4-1 zu finden. Im Folgenden wird die Paretomenge im Bildraum zur besseren Unterscheidung als Paretofront bezeichnet. In Bild 4-1 ist ebenfalls die geometrische Bedeutung des Vektors γ angedeutet. Dieser steht immer senkrecht

auf der Tangentialebene im zugehörigen Paretopunkt $F(p^*)$. Dies folgt aus der Betrachtung der skalaren Funktion

$$g_\gamma(p) = \sum_{i=1}^{n_o} \gamma_i f_i(p) = \gamma^T F(p), \text{ mit } \gamma_i \geq 0, (1 \leq i \leq n_o).$$

Die notwendige Bedingung für ein Minimum von $g_\gamma(p)$ führt direkt auf die KKT-Bedingung, falls γ entsprechend normiert ist. Zudem wird durch $\gamma^T F(p) = c$ eine Ebene im Bildraum beschrieben, zu der γ orthogonal ist. Die Minimierung von g_γ entspricht geometrisch einer Verschiebung dieser Ebene in Richtung des Koordinatenursprungs. Das Minimum ergibt sich für den Paretopunkt, an dem die Ebene tangential an der Paretomenge liegt.

Zur Berechnung der Paretomenge wird das MOP (4-1) oftmals in verschiedene skalare Optimierungsprobleme umgewandelt, die in geeigneter Weise durch einen Satz von Parametern beschrieben sind. Durch die Lösung der skalaren Optimierungsprobleme werden einzelne Paretopunkte bestimmt. Über Variationen der Parametrierung kann eine Approximation der gesamten Paretomenge erzielt werden. Der Vorteil dieser Herangehensweise liegt darin, dass die Algorithmen und theoretischen Resultate, wie beispielsweise notwendige und hinreichende Optimalitätsbedingungen, aus der skalaren Optimierung genutzt werden können.

Ein Beispiel für die Umwandlung eines MOP in mehrere skalare Optimierungsprobleme ist die bereits erwähnte Funktion $g_\gamma(p)$. Mit diesem Ansatz, bekannt als gewichtete Summe, können durch eine Variation der Parametrierung γ verschiedene Paretopunkte berechnet werden.

Ein weiteres Beispiel ist die sogenannte ε -Constraint Methode. Hier wird nur eine einzelne Zielfunktion minimiert. Für die übrigen Zielfunktionen werden obere Schranken festgelegt, die in Form von Gleichheits- oder Ungleichheitsnebenbedingungen in das skalare Optimierungsproblem einfließen. Die Parametrierung wird bei diesem Verfahren von den oberen Schranken gebildet.

Als letztes Beispiel sei noch die NBI Methode (Normal Boundary Intersection Method) genannt, bei der ein geometrischer Ansatz verfolgt wird. Zu Beginn werden die Minima der einzelnen Zielfunktionen berechnet. Anschließend wird zwischen diesen Punkten im Bildraum ein Simplex definiert. Ausgehend von Punkten auf diesem Simplex werden in orthogonaler Richtung Schnittpunkte mit der Paretofront durch die Lösung eines speziellen Optimierungsproblems berechnet. Weitere Details zu den drei erwähnten Beispielen sind z. B. in [Hil01] beschrieben.

Die Umwandlung eines MOP in mehrere skalare Optimierungsprobleme hat zwei Nachteile. Zum einen ist nicht gesichert, dass die berechneten Lösungen wirklich paretooptimale Punkte des MOP sind. Oftmals werden lokale Minima berechnet, die von den tatsächlichen Paretopunkten dominiert werden. Zum anderen ist

gerade aus Sicht des Anwenders eine gleichmäßige Verteilung der Paretopunkte erwünscht. Diese lässt sich aber a priori meist nicht vorhersehen und daher oftmals nur iterativ herstellen. Betrachtungen zu dieser Problematik sind u. a. in [Mün12] zu finden.

Als Alternative zu den Skalarisierungstechniken können Verfahren eingesetzt werden, mit denen die gesamte Paretomenge approximiert wird. Weit verbreitet ist auf diesem Gebiet der Einsatz stochastischer Methoden. Ein Beispiel stellen die evolutionären Algorithmen dar, die durch eine stetige Verbesserung einer Population, d. h. einer Menge von Vektoren im Parameterraum, die Paretomenge iterativ approximieren, siehe z. B. [Deb99].

In der vorliegenden Arbeit wird eine weitere Variante, ein mengenorientierter Ansatz, verwendet, um Paretomengen zu berechnen. Im Folgenden wird ein kurzer Überblick über die Grundlagen dieses Optimierungsansatzes gegeben. Die Grundidee besteht darin, eine definierte Anfangsmenge, beschrieben durch ein mehrdimensionales Rechteck im \mathbb{R}^{n_p} , bezeichnet als eine Box \mathcal{B}_0 , sukzessive zu unterteilen und nur die Boxen zu behalten, in denen Teile der Paretomenge liegen. Die hierfür verwendeten Algorithmen wurden ursprünglich zur Berechnung relativer, globaler Attraktoren dynamischer Systeme entwickelt [DH97]. Implementiert sind sie in dem Softwaretool GAIO [DFJ00], dessen spezielle Erweiterungen auf MOP innerhalb der vorliegenden Arbeit genutzt werden. Eine Übertragung der Methodik aus dem Bereich der dynamischen Systeme auf die Mehrzieloptimierung ist möglich, da die zu berechnenden Paretomengen relative, globale Attraktoren spezieller dynamischer Systeme darstellen, was im Weiteren noch gezeigt wird.

Ein globaler Attraktor eines dynamischen Systems stellt eine attraktive, invariante Teilmenge des Zustandsraums dar. Dies bedeutet, dass sich Trajektorien mit beliebigen Anfangswerten an den Attraktor annähern. Ein relativer globaler Attraktor erweitert den Begriff des Attraktors in der Hinsicht, dass man zunächst eine kompakte Teilmenge Q des Zustandsraums definiert und untersucht, ob es für alle Punkte aus Q eine attraktive Teilmenge in Q gibt. Ist das dynamische System in diskreter Form beschrieben durch

$$x_{k+1} = \mathcal{F}(x_k) \text{ mit } k \in \mathbb{N}_0,$$

so ist der relative globale Attraktor gegeben als

$$A_Q = \bigcap_{k \geq 0} \mathcal{F}^k(Q).$$

Ein relativer, globaler Attraktor entspricht im Allgemeinen nicht dem Anteil des globalen Attraktors an der Ausgangsmenge Q . Umgekehrt ist der relative, globale Attraktor jedoch stets eine Teilmenge des globalen Attraktors.

Der Unterteilungsalgorithmus zur Approximation eines relativen, globalen Attraktors besteht aus der wiederholten Ausführung der folgenden beiden Schritte, ausgehend von einer Startbox $\mathcal{B}_0 = Q$ mit einem bestimmten Durchmesser $\text{diam}(\mathcal{B}_0)$.

- (i) Erzeuge eine feinere Boxüberdeckung $\hat{\mathcal{B}}_k$ mit

$$\bigcup_{B \in \hat{\mathcal{B}}_k} B = \bigcup_{B \in \mathcal{B}_{k-1}} B$$

und $\text{diam}(\hat{\mathcal{B}}_k) = \theta_k \text{diam}(\mathcal{B}_{k-1})$ mit $0 < \theta_k < 1$.

- (ii) Bestimme die neue Boxüberdeckung als

$$\mathcal{B}_k = \left\{ B \in \hat{\mathcal{B}}_k : \exists \hat{B} \in \hat{\mathcal{B}}_k \text{ mit } \mathcal{F}^{-1}(B) \cap \hat{B} \neq \emptyset \right\},$$

d. h. es werden nur diejenigen Boxen behalten, in die Trajektorien „hineinlaufen“.

Falls die Funktion \mathcal{F} des untersuchten dynamischen Systems ein Diffeomorphismus ist, d. h. eine stetig differenzierbare und bijektive Abbildung, so konvergiert die Boxüberdeckung \mathcal{B}_k für $k \rightarrow \infty$ gegen den relativen, globalen Attraktor, siehe [DH97]. Bei der numerischen Umsetzung wird dabei die Auswahl in Schritt (ii) durch eine endliche Zahl von Testpunkten realisiert.

Ein Zusammenhang mit der Mehrzieloptimierung kann hergestellt werden, indem erneut die KKT-Bedingung benutzt wird. Dazu betrachtet man die Funktion

$$\kappa : \mathbb{R}^{n_p} \rightarrow \mathbb{R}^{n_o}, \quad \kappa(p) = \sum_{i=1}^{n_o} \hat{\gamma}_i \nabla f_i(p),$$

wobei $\hat{\gamma}$ für jedes feste $p \in \mathbb{R}^{n_p}$ eine Lösung des quadratischen Minimierungsproblems

$$\min_{\gamma \in \mathbb{R}^{n_o}} \left\{ \left\| \sum_{i=1}^{n_o} \gamma_i \nabla f_i(p) \right\|_2^2 ; \gamma_i \geq 0, i = 1, \dots, n_o, \sum_{i=1}^{n_o} \gamma_i = 1 \right\}$$

ist. Die Paretopunkte des MOP bilden Nullstellen von κ , da in den Nullstellen die KKT-Bedingung erfüllt ist. Darüber hinaus stellt $\kappa(p)$ außerhalb der Nullstellen stets eine Abstiegsrichtung des MOP dar. Durch eine geeignete Formulierung diskreter dynamischer Systeme der Form

$$p_{k+1} = p_k + h_k \kappa(p_k), \tag{4-2}$$

mit einer Schrittweite h_k , lässt sich zeigen, dass sämtliche Trajektorien gegen die Nullstellenmenge von κ konvergieren, die somit den globalen Attraktor bildet. Der mathematische Beweis sowie eine ausführliche Beschreibung dieses mengenorientierten Ansatzes sind in [Sch04] zu finden.

Bei der praktischen Anwendung hängen die Konvergenz und die Konvergenzgeschwindigkeit stark von dem gegebenen Optimierungsproblem ab. Insbesondere die Anzahl der verwendeten Testpunkte innerhalb der Boxstruktur hat einen entscheidenden Einfluss sowohl auf die Laufzeit als auch auf die Güte der Approximation der Paretomenge. In ungünstigen Fällen, z. B. bei langsamer Konvergenz oder hochdimensionalen Parameterräumen, kann die Anzahl der Boxen stark ansteigen, was wiederum zu langen und speicheraufwändigen Berechnungen führt. Abhilfe können hier Varianten des mengenorientierten Ansatzes schaffen, die eine Boxüberdeckung der Paretofront im Bildraum bestimmen, siehe [Del08].

Eine andere Alternative zur Lösung des letztgenannten Problems besteht in dem Einsatz eines Unterteilungsalgorithmus, der keine Ableitungen benötigt. Eine derartige Variante, bezeichnet als *sampling*-Algorithmus, ist ebenfalls in GAIO verfügbar und wurde erfolgreich im Rahmen dieser Arbeit eingesetzt. Beim *sampling*-Algorithmus wird der zweite Schritt des Unterteilungsalgorithmus durch einen Nichtdominanztest ersetzt. Dies bedeutet, dass erneut in jeder Box Testpunkte generiert werden. Aber statt für jeden dieser Testpunkte eine Trajektorie für das System (4-2) bzgl. κ zu berechnen, wird jeder Punkt in den Bildraum unter Anwendung von F abgebildet. Aus den resultierenden Punkten im Bildraum werden die nichtdominierten Punkte bestimmt. In der Boxüberdeckung werden daraufhin alle Boxen entfernt, die keine nichtdominierten Punkte enthalten. Zusätzlich werden in einem Archiv alle nichtdominierten Punkte abgelegt, sodass während der Unterteilungsschritte keine bereits bestimmten Punkte verloren gehen.

Der klare Vorteil des mengenorientierten Ansatzes liegt in seiner Globalität. Insbesondere beim *sampling*-Algorithmus erhält der Anwender als Ergebnis eine Menge nichtdominierter Punkte, die aus zufällig gewählten und über den gesamten Parameterraum verteilten Testpunkten generiert wurde, sodass lokale Minima vermieden werden. Ein weiterer Vorteil ist die gleichmäßige Verteilung der berechneten Paretopunkte bei dem mengenorientierten Ansatz. Diese ergibt sich direkt aus der Verfeinerung der Boxen, da jede Box mindestens einen der berechneten Paretopunkte enthält.

4.2 Hierarchische Mehrzieloptimierung

Die hierarchische Optimierung dient dazu, mehrere voneinander abhängige MOP effizient zu lösen. Die beteiligten MOP sind nicht gleichberechtigt, sondern bilden eine hierarchische Struktur. Solch eine Struktur ergibt sich etwa aus der in Kapitel

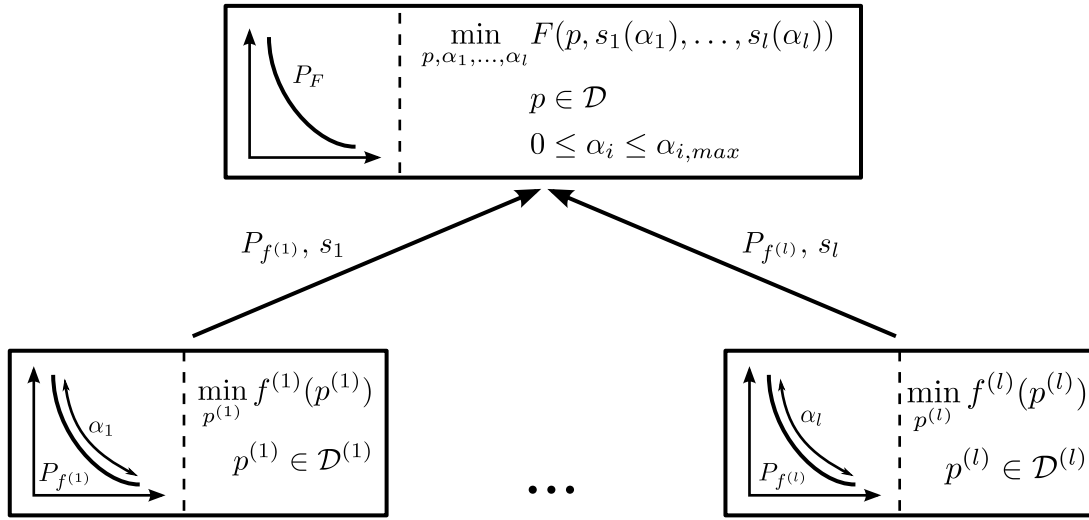


Bild 4-2: Schematische Darstellung der hierarchischen Optimierung für zwei Hierarchieebenen

1 beschriebenen hierarchischen Modellierung selbstoptimierender Systeme, indem für jedes Teilsystem eigene Zielfunktionen und Optimierungsparameter definiert werden. Die Gesamtheit der Optimierungsprobleme besitzt dann die gleiche hierarchische Struktur wie das selbstoptimierende System.

In [Del08] werden sogenannte Bi-Level MOP betrachtet, die für den Spezialfall einer Hierarchie aus zwei Ebenen mit genau einem unterlagerten und einem überlagerten MOP entwickelt wurden. Für diese spezielle Art der hierarchischen Optimierung werden in [Del08] sowohl theoretische Optimalitätsbedingungen, als auch auf die Lösung von Bi-Level MOPs zugeschnittene Algorithmen vorgestellt.

Innerhalb dieser Arbeit wird ein allgemeinerer Ansatz verfolgt, der mehrere unterlagerte MOP und eine beliebige Anzahl von Hierarchieebenen zulässt. Als Voraussetzung müssen jedoch die Optimierungsparameter der MOP auf gleicher Hierarchieebene disjunkt zueinander sein. Stammt die Hierarchie aus der Systemstruktur des selbstoptimierenden Systems, wird dies oftmals der Fall sein. Das im folgenden beschriebene Verfahren wurde bereits mehrfach im Kontext selbstoptimierender Systeme eingesetzt, siehe [MAK⁺08, KWTD11, HKK⁺12, Mün12].

Die prinzipielle Funktionsweise der hierarchischen Optimierung ist in Bild 4-2 dargestellt. Betrachtet werden auf einer nicht näher bestimmten Hierarchieebene die unterlagerten MOPs

$$\min_{p^{(i)}} \{ f^{(i)}(p^{(i)}) : p^{(i)} \in \mathcal{D}^{(i)} \subseteq \mathbb{R}^{n_{p,i}} \}$$

mit den Zielfunktionen $f^{(i)} : \mathbb{R}^{n_{p,i}} \rightarrow \mathbb{R}^{n_{o,i}}$ ($1 \leq i \leq l$), wobei innerhalb dieser Arbeit einfachheitshalber nur unterlagerte MOP mit zwei Zielfunktionen ($n_{o,i} = 2$) betrachtet werden. Die Verallgemeinerung auf mehr als zwei Zielfunktionen ist jedoch möglich, siehe Abschnitt 5.5.3. Die zugehörigen Paretomengen seien mit $P_{f^{(i)}} \subseteq \mathbb{R}^{n_{p,i}}$ bezeichnet. Das überlagerte MOP, das auch von den Parametern der unterlagerten MOP abhängt, wird nun in der Weise eingeschränkt, dass nur paretooptimale Konfigurationen der unterlagerten MOP verwendet werden. Somit ergibt sich für das überlagerte MOP

$$\min_{p, p^{(1)}, \dots, p^{(l)}} \{F(p, p^{(1)}, \dots, p^{(l)}) : p \in \mathcal{D}, p^{(i)} \in P_{f^{(i)}}\}.$$

Die Lösung des hierarchischen Optimierungsproblems erfolgt nun „bottom-up“ in mehreren Schritten. Zunächst werden die unterlagerten MOP gelöst und die zugehörigen Paretomengen $P_{f^{(i)}}$ berechnet. Unter der Annahme, dass diese MOP hinreichend glatt sind, stellen die Paretomengen eindimensionale Kurven (Mannigfaltigkeiten) im jeweiligen Parameterraum dar. Daher kann eine Parametrisierung über skalare Parameter α_i mittels der Funktionen

$$s_i : [0, \alpha_{i,max}] \rightarrow P_{f^{(i)}}, s_i(\alpha_i) = p^{*(i)}, (1 \leq i \leq l)$$

berechnet werden. Konkrete Varianten derartiger Parametrisierungen werden ausführlich in Kapitel 5 behandelt. Das überlagerte Optimierungsproblem kann nun bezüglich α_i ausgedrückt werden als

$$\min_{p, \alpha_1, \dots, \alpha_l} \{F(p, s_1(\alpha_1), \dots, s_l(\alpha_l)) : p \in \mathcal{D}, 0 \leq \alpha_i \leq \alpha_{i,max}\}$$

und mit den in Abschnitt 4.1 beschriebenen Verfahren der Mehrzieloptimierung gelöst werden. Es unterscheidet sich qualitativ nicht mehr von den unterlagerten MOP und kann daher auch selbst wieder ein unterlagertes MOP bilden.

Die hierarchische Optimierung besitzt zwei Vorteile verglichen mit der Lösung eines gesamten MOP, in dem sämtliche Zielfunktionen der gesamten Hierarchie unter Variation aller Optimierungsparameter berücksichtigt werden. Zum einen lassen sich die Teilprobleme wesentlich schneller lösen, da sowohl weniger Zielfunktionen als auch weniger Optimierungsparameter in jedem Teilproblem berücksichtigt werden müssen. Zum anderen ist eine hochdimensionale Paretofront aus technischer Sicht nur sehr schwer analysierbar und interpretierbar. Geeignete Parameterwerte für konkrete Aufgabenstellungen lassen sich nur schlecht bestimmen. Dem gegenüber sichert der hierarchische Ansatz ab, dass alle Parameter der unterlagerten Teilprobleme optimal bezüglich der dort genutzten Ziele sind und dennoch eine gewisse Variabilität aufweisen, um das Verhalten des Gesamtsystems zu optimieren.

Durch die Einschränkung der unterlagerten MOP kann jedoch die Paretooptimalität bezüglich eines nicht beschränkten Gesamtproblems verloren gehen. Die mit der hierarchischen Optimierung berechneten Paretomengen können bestenfalls gleiche Zielfunktionswerte aufweisen, werden im Allgemeinen aber aufgrund der Parameterbeschränkung von den theoretisch erzielbaren Ergebnissen abweichen, siehe auch [MAK⁺08] für einen Vergleich der beiden Ansätze. Dies stellt einen Nachteil der hierarchischen Optimierung dar. Zudem müssen, wie bereits erwähnt, die unterlagerten MOP stets unabhängig voneinander formuliert werden können, damit der hier aufgezeigte Lösungsansatz angewendet werden kann.

4.3 Optimierungsmodell für selbstoptimierende Systeme

Die in den vorhergehenden Abschnitten beschriebenen Verfahren der Mehrzieloptimierung werden im Rahmen selbstoptimierender Systeme zur Berechnung optimaler Systemkonfigurationen eingesetzt. Die Anwendung von Mehrzieloptimierung besitzt hier entscheidende Vorteile. Zum einen können mehrere innerhalb des Entwurfsprozesses definierte Ziele gleichzeitig betrachtet werden. Zum anderen steht mit der Paretomenge als Ergebnis eine Menge optimaler Kompromisse zur Verfügung. Werden die zulässigen Systemkonfigurationen auf die Paretomenge beschränkt, so agiert das System, zumindest der Theorie nach, stets optimal. Die für die Realisierung der Selbstoptimierung notwendige Anpassungsfähigkeit des Systems bleibt erhalten, indem das System zur Laufzeit den am besten geeigneten Punkt auf der Paretomenge auswählt.

Die Optimierungsprobleme im Kontext selbstoptimierender Systeme zielen stets auf eine Verbesserung des dynamischen Verhaltens ab. Da sich das selbstoptimierende System während des Betriebs an veränderliche Situationen anpassen soll, handelt es sich nicht um eine Optimierung der Komponenten, z. B. hinsichtlich ihrer Geometrie oder Dimensionierung. Derartige Systemparameter müssen vor der Fertigung des Systems in geeigneter Weise ausgelegt werden (vgl. [VDI04b]) und können zur Laufzeit nur noch in sehr eingeschränkter Weise oder gar nicht mehr variiert werden.

Im Gegensatz hierzu lässt sich die Informationsverarbeitung des Systems auch zur Laufzeit noch adaptieren. Hier hat die implementierte Regelung den entscheidenden Einfluss auf das dynamische Verhalten. Aus diesem Grund stellen die Parameter der Regelung in den auf selbstoptimierende Systeme zugeschnittenen Optimierungsproblemen die Optimierungsparameter dar. Die Art der Parameter der Regelung ist hierbei nicht näher eingeschränkt. Neben Verstärkungsfaktoren aus klassischen Reglern für Eingrößensysteme sind auch Gewichtungen z. B. aus dem Riccati-Entwurf oder andere abstrakte Parameter vorstellbar, die beispielsweise eine Störgrößenkompensation beschreiben. Aus diesem Grund handelt es sich um eine Verallgemeinerung optimaler Regelungen, wie der LQ-Regelung

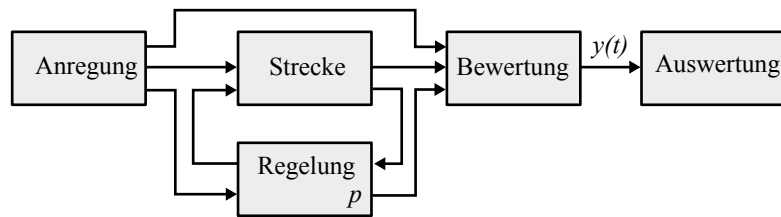


Bild 4-3: Optimierungsmodell für die simulationsbasierte Mehrzieloptimierung selbstoptimierender Systeme

[AM71] oder auch der modellprädiktiven Regelung [GP11, Gra12], die zumindest in ihrer ursprünglichen Form feste Ziele und bestimmte Optimierungsparameter vorgeben.

Eine Optimierung des dynamischen Verhaltens in der geschilderten allgemeinen Form kann nur mit Hilfe eines geeigneten Modells des selbstoptimierenden Systems durchgeführt werden. Die Erstellung eines verifizierten und validierten Modells der Regelstrecke stellt selbst ein komplexes Problem mit vielen unterschiedlichen Facetten dar, auf das an dieser Stelle nicht näher eingegangen wird. In der Literatur findet sich eine große Anzahl von Arbeiten, die sich mit der Modellbildung beschäftigen. Beispielhaft sei auf [Ise08, SW10, HPZA10, Jan10] verwiesen.

Innerhalb dieses Abschnitts wird auch keine Unterscheidung getroffen, ob das Modell der geregelten Strecke direkt aus physikalischen Gesetzen abgeleitet wurde oder Teil des in Kapitel 1 beschriebenen hierarchischen Modells ist und somit wiederum separat geregelte und optimierte Teilsysteme enthält. Für die folgenden Ausführungen wird davon ausgegangen, dass sowohl die Regelstrecke, als auch die Regelung in geeigneter Weise durch ein Modell gegeben sind. Es ist an dieser Stelle ebenfalls nicht erheblich, ob die Modelle linear oder nichtlinear sind. Die Mehrzieloptimierung kann ohne Weiteres auch auf nichtlineare Systeme angewandt werden. Ausgehend von dem vorhandenen Modell der geregelten Strecke wird im Folgenden dargestellt, wie ein Optimierungsmodell für die Mehrzieloptimierung selbstoptimierender Systeme aufgebaut werden kann.

Die Zielgrößen, bzgl. der das System optimiert werden soll, sind im Allgemeinen nicht direkt aus dem Modell ablesbar. Die Ausnahme bilden sehr einfache Ziele wie beispielsweise die Dämpfung bestimmter Eigenfrequenzen, die mit Hilfe von Bode-Diagrammen analysiert werden kann. Allgemein ist es notwendig, bestimmte, auf die Zielfunktionen abgestimmte Simulationen des geregelten Systems durchzuführen. Abgesehen von dem Fall, dass das Systemverhalten für spezielle und vorher bekannte Situationen optimiert werden soll, führt man zumeist möglichst allgemeingültige Referenzsimulationen aus. In diesen müssen sich alle relevanten dynamischen Eigenschaften des Systems abzeichnen. Für solch eine

Simulation wird ein erweitertes Modell, dargestellt in Bild 4-3, verwendet. Dieses enthält neben der Regelstrecke und der Regelung die Auswertung der eigentlichen Zielfunktionen sowie ein Anregungs- und ein Bewertungsmodell, mit deren Hilfe das Optimierungsproblem spezifiziert wird.

Der Regelungsblock in Bild 4-3 ist hier sehr allgemein zu verstehen und nicht auf klassische Rückführungen begrenzt. Es können prinzipiell auch Regelkonzepte auf Basis der sogenannten Zwei-Freiheitsgrade-Struktur [Rop09] zum Einsatz kommen, die neben einer Rückführung auch Anteile zur Steuerung enthalten. Ebenso sind auch Zustands- oder Störgrößenbeobachter sowie eine Störgrößenaufschaltung denkbar und mit den Methoden der Mehrzieloptimierung behandelbar. Die einzige Voraussetzung besteht darin, dass eine feste Anzahl kontinuierlicher Parameter, im Weiteren bezeichnet durch den Vektor $p \in \mathbb{R}^{n_p}$, definiert werden kann. Diese Parameter müssen die gewünschten Freiheitsgrade des Regelungsentwurfs widerspiegeln.

Im Anregungsmodell werden die Störgrößen und falls erforderlich auch die Führungsgrößen berechnet. Hierdurch wird die Situation bzw. das Szenario definiert bzgl. dessen das dynamische Verhalten optimiert werden soll. Als Führungsgrößen sind generische Signale wie beispielsweise Sprung- oder Rampenfunktionen oder auch Steuertrajektorien für das System denkbar. Die Störgrößen werden oftmals durch stochastische Signale nachgebildet. Im Rahmen der in dieser Arbeit durchgeführten Optimierungen wird zum Beispiel ein Gauß'sches weißes Rauschen verwendet. In besonderen Fällen sind die Anregungsdaten genauer bekannt. Beispielsweise wurden bei der aktiven Spurführung des RailCab die Anregungen mit Hilfe iterativer Lernverfahren bestimmt und anschließend eine Mehrzieloptimierung durchgeführt [GWTD08]. Allgemein betrachtet wirken die Störgrößen direkt auf die Regelstrecke, während die Führungsgrößen einen Eingang der Regelung darstellen. Eine Ausnahme bildet hier die Störgrößenaufschaltung, bei der die messbaren Störgrößen auf den Regelungsblock wirken.

Das Bewertungsmodell definiert zusammen mit dem Auswertungsblock die eigentlichen Ziele bzgl. der das System optimiert werden soll. In vorhergehenden, verwandten Arbeiten, z.B. [Mün12], [Kas85], wurden diese beiden Teile als Einheit betrachtet. Zusammen dienen sie der Aufbereitung der Systemausgänge und der Berechnung des Zielfunktionswertes. Innerhalb dieser Arbeit beinhaltet das Bewertungsmodell alle Komponenten dieser Aufbereitung, die über eine eigene Dynamik verfügen. Hierunter sind u. a. Frequenzgewichtungen zu verstehen wie sie etwa in der VDI-Norm [VDI04a] zur Beschreibung der Auswirkungen von Schwingungen auf das menschliche Empfinden angegeben sind. In das Bewertungsmodell können neben bestimmten Ausgängen der Regelstrecke, die nicht unbedingt den Messgrößen entsprechen müssen, auch Signale aus der Regelung oder dem Anregungsmodell einfließen. Das Anregungsmodell spielt beispielswei-

se eine Rolle, wenn zur Auswertung der Zielfunktionen ein Soll-Istwert Vergleich zwischen Führungs- und Regelgröße notwendig ist. Ein Ausgang der Regelung kann etwa zur Bestimmung des Regelaufwands ähnlich wie beim Riccati-Entwurf genutzt werden.

Der Auswertungsblock beschreibt die eigentliche Berechnung des numerischen Zielfunktionswertes aus den Zeitsignalen des Simulationsmodells. Hier können sehr unterschiedliche Funktionen verwendet werden, beispielsweise Mittelwerte, allgemeine Integrale von Zeitsignalen oder auch einzelne Werte zu bestimmten Zeiten wie das Maximum oder Minimum eines gewissen Signals. Eine Übersicht über mögliche Varianten für den Einsatz im Rahmen selbstoptimierender Systeme ist in [Mün12] zu finden. Im Hinblick auf die in Abschnitt 4.1 beschriebenen Eigenschaften von MOP lassen sich einige generelle Anforderungen ableiten.

Nach Möglichkeit sollten die Zielfunktionen zweimal stetig differenzierbar sein. Damit dies erfüllt ist, müssen nicht nur die Funktionen im Block Auswertung, sondern auch der Ausgang $y(t)$ des Bewertungsblocks hinreichend glatt sein. Hierbei ist mitunter die Kombination beider Anteile des Optimierungsmodells von Bedeutung. So ist es beispielsweise ausreichend, wenn y einmal stetig differenzierbar ist, falls zur Auswertung das Integral $\int y$ verwendet wird. Bei der Definition des Optimierungsproblems sollten daher sprunghafte Änderungen des Systemverhaltens in der Nähe des Ausgangs y bei Variation der Parameter p vermieden werden.

Zur Vermeidung numerischer Probleme bei der Lösung des MOP ist zudem auf eine angemessene Skalierung der Zielfunktionen zu achten, sodass die Wertebereiche aller Zielfunktionen in der gleichen Größenordnung liegen. Generelle Hinweise hierzu sind bspw. in [Mie99] zu finden.

Um sicherzustellen, dass alle berechneten Paretopunkte globale und nicht etwa lokale Paretopunkte darstellen, ist ein konvexes MOP hinreichend. Die Konvexität der gesamten Zielfunktion ist bei der simulationsbasierten Mehrzieloptimierung nur schwer zu prüfen, da in vielen Fällen keine analytische Beschreibung der Zielfunktionen vorliegt. Umgekehrt lässt sich aber schließen, dass das gesamte MOP nicht konvex sein kann, wenn im Auswertungsblock nichtkonvexe Funktionen verwendet werden. Die Konvexität geht zudem verloren, wenn einzelne Optimierungsparameter keinerlei Einfluss auf eine der Zielfunktionen haben. Diese beiden Aspekte sind daher nach Möglichkeit bei der Definition des MOP zu vermeiden.

4.4 Anwendungen der Mehrzieloptimierung

Der folgende Abschnitt veranschaulicht die beschriebenen Konzepte und Methoden der Mehrzieloptimierung. Dazu werden zwei verschiedene Mehrzieloptimierungsprobleme definiert und die zugehörigen Lösungen präsentiert. Beide Anwen-

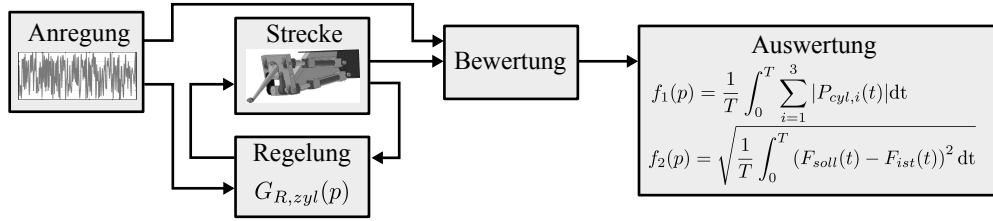


Bild 4-4: Optimierungsmodell des Aktormoduls

dungsbeispiele beschäftigen sich mit dem Feder-Neige-Prüfstand. In Abschnitt 4.4.1 wird die Regelung eines einzelnen Aktormoduls betrachtet und optimiert. Anschließend wird in Abschnitt 4.4.2 der gesamte Feder-Neige-Prüfstand behandelt. Beide Systeme wurden bereits in Abschnitt 1.3 vorgestellt. Die hier gezeigten Ergebnisse dienen darüber hinaus als Anwendungsbeispiel für die im nächsten Kapitel vorgestellte Variante der parametrischen Modellordnungsreduktion.

4.4.1 Mehrzieloptimierung des Aktormoduls

Die Funktion der Aktormodule besteht darin, die Federfußpunkte der GFK-Federn in geeigneter Weise zu verstellen, sodass eine vorgegebene Kraft zwischen der Aufbaumasse und der Anregungseinheit vorliegt. Dazu werden über eine inverse Kinematik aus den von der Aufbauregelung vorgegebenen Sollkräften die benötigten Sollpositionen der drei Hydraulikzylinder berechnet.

Jeder der drei Hydraulikzylinder verfügt über eine eigene Positionsregelung des Zylinderkolbens, die durch einen PD-Regler mit der Übertragungsfunktion

$$G_{R,zyl,i}(s) = K_{p,i} + \frac{K_d s}{T_d s + 1}$$

realisiert wird. In den Vorarbeiten [Krü09] und [Sch09] hat sich gezeigt, dass der Einfluss von K_d und T_d auf das dynamische Verhalten des Gesamtsystems gering ist. Daher wird für diese beiden Faktoren im Folgenden ein konstanter Wert verwendet. Der wesentliche Einfluss auf das dynamische Verhalten der Aktormodule geht von den proportionalen Anteilen, beschrieben durch die drei Verstärkungsfaktoren $K_{p,1}$, $K_{p,2}$ und $K_{p,3}$, aus. Diese stellen daher die Optimierungsparameter p der innerhalb dieses Abschnitts beschriebenen Mehrzieloptimierung dar. Das zugehörige Optimierungsmodell ist in Bild 4-4 dargestellt und wird im Folgenden näher beschrieben.

Die erste betrachtete Zielgröße stellt gemittelte Leistung dar, die bei der Verstellung der Hydraulikzylinder anfällt. Die hydraulische Leistung, die während des Betriebs zur Verfügung gestellt werden muss, ergibt sich als Produkt

$$P_{hyd} = p_{sup} \cdot Q_{sup}(t),$$

wobei p_{sup} den als konstant angenommenen Versorgungsdruck und $Q_{sup}(t)$ den aus der Versorgungseinheit kommenden Volumenstrom darstellen. In dem betrachteten Optimierungsmodell wird der Volumenstrom über die momentane Geschwindigkeit $v_{zyl}(t)$ des Zylinderkolbens multipliziert mit der Kolbenfläche angenähert. Da es sich bei den Hydraulikzylindern um Differentialzylinder handelt, müssen die beiden unterschiedlichen Kolbenflächen A_1 und A_2 der beiden Zylinderkammern berücksichtigt werden. Daraus ergibt sich die vom Vorzeichen der Zylindergeschwindigkeit abhängige Näherung für die hydraulische Leistung

$$P_{zyl,i}(t) = \begin{cases} A_1 p_{sup} v_{zyl,i}(t) & \text{für } v_{zyl,i}(t) > 0, \\ A_2 p_{sup} v_{zyl,i}(t) & \text{für } v_{zyl,i}(t) < 0, \end{cases}$$

wobei das Vorzeichen angibt, in welche Kammer der Volumenstrom gerichtet ist. Weitere Details zu dieser Näherung sind in [Mün12] zu finden.

Als Zielfunktion werden die Leistungen der drei Hydraulikzylinder summiert und über die Zeit mit Hilfe eines Integrals gemittelt, woraus sich die Funktion

$$f_1(p) = \frac{1}{T} \int_0^T \sum_{i=1}^3 |P_{zyl,i}(t)| dt \quad (4-3)$$

ergibt. Die Zeit T gibt die Simulationszeit des Modells an und bleibt konstant während der gesamten Optimierung.

Das zweite Ziel stellt die Regelgüte dar. Diese kann mathematisch durch die Abweichung der Ist-Kraft $F_{ist}(t)$ in der GFK-Feder von der Soll-Vorgabe $F_{soll}(t)$ beschrieben werden. Hierbei wird die durchschnittliche quadratische Abweichung

$$f_2(p) = \sqrt{\frac{1}{T} \int_0^T (F_{soll}(t) - F_{ist}(t))^2 dt} \quad (4-4)$$

betrachtet.

Das Modell der Regelstrecke innerhalb des Optimierungsmodells besteht aus einem nichtlinearen Modell des Aktormoduls. Dieses beinhaltet ein nichtlineares MKS-Modell der Lenkerplatte sowie nichtlineare Modelle der Hydraulikzylinder. Das Aktormodul sowie der nicht mit diesem verbundene zweite Ankoppelpunkt der GFK-Feder werden fest mit dem Inertialsystem verbunden

Als Anregungsmodell wird ein stochastisches Signal verwendet. Es handelt sich um ein bandbegrenztes weißes Rauschen, das mit Hilfe eines PT1-Glieds geglättet wird. Ein spezielles Bewertungsmodell wird für das Aktormodul nicht benötigt. Die Ausgänge des Streckenmodells werden ohne zusätzliche Dynamik über die Zielfunktionen (4-3) und (4-4) ausgewertet.

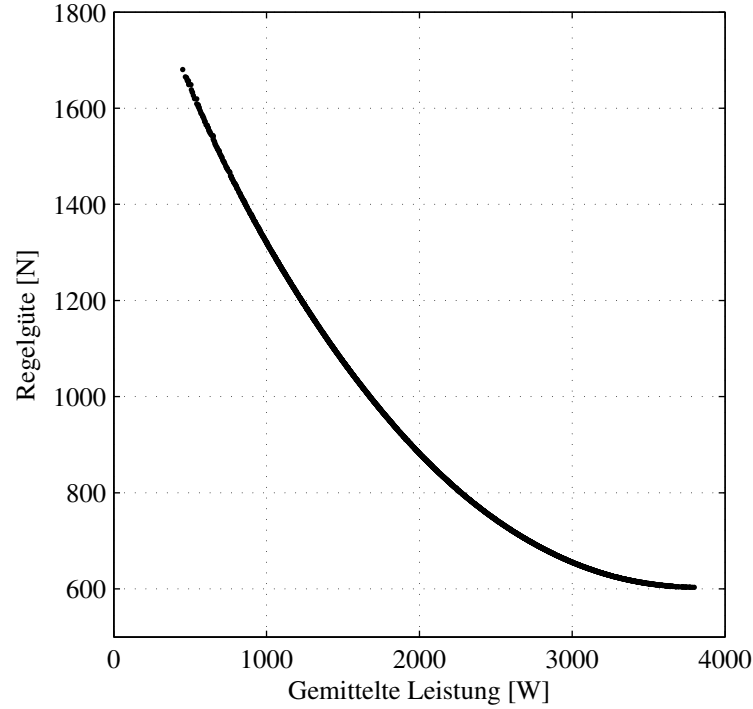


Bild 4-5: Paretofront des Aktormoduls

Das resultierende MOP mit den Optimierungsparametern $p = [K_{p,1}, K_{p,2}, K_{p,3}]$ ergibt sich als

$$\min_p \{ [f_1(p), f_2(p)] : p \in \mathcal{S} \subseteq \mathbb{R}^3 \}.$$

Als Definitionsbereich \mathcal{D} werden für die drei Verstärkungsfaktoren $p_{i,\min} = 0$ und $p_{i,\max} = 3000$ als untere bzw. obere Schranke definiert. Das MOP wird mit den Unterteilungsalgorithmen aus Abschnitt 4.1 gelöst. Die resultierende Paretofront ist in Bild 4-5 zu finden. Es ist deutlich zu erkennen, dass es sich um gegenläufige Zielfunktionen handelt, die kein gemeinsames Minimum besitzen. Dies entspricht auch der physikalischen Anschauung, da eine höhere Regelgüte mehr Energieaufwand erfordert.

Die Paretomenge im Urbildraum ist Bild 4-6 zu entnehmen. Der rechte Teil zeigt die Boxüberdeckung, die mit GAIO berechnet wurde, der linke Teil die nichtdominierten Testpunkte, die sich ausschließlich innerhalb der Boxen befinden. Obwohl es sich noch um ein dreidimensionales Objekt handelt, stellt es doch eine gute Annäherung an die theoretisch zu erwartende, eindimensionale Paretomenge dar.

Im Rahmen der Optimierung wurden insgesamt 17 Unterteilungsschritte durchgeführt, in denen alle Boxen in wechselnden Dimensionen halbiert wurden. Weitere

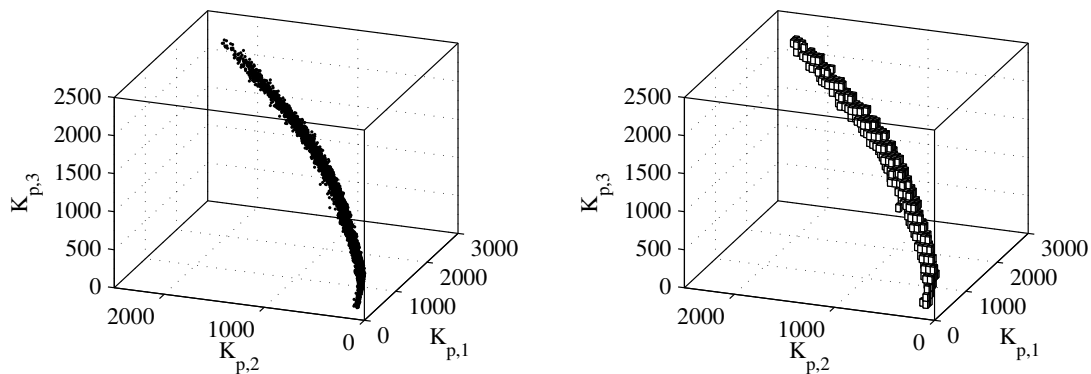


Bild 4-6: Paretomenge und Boxüberdeckung des Aktormoduls

Unterteilungen könnten prinzipiell zu einer weiteren Annäherung an die theoretische Paretomenge führen. Allerdings ist die Approximationsgüte beschränkt, da aufgrund der ausgeführten Simulation numerische Integrationsfehler in der Auswertung der Zielfunktion vorhanden sind. Diese führen zu einer Art numerischem Rauschen und beschränken die Berechnungsgenauigkeit der Zielfunktionswerte.

Die berechnete Paretomenge entspricht ebenfalls den zu erwartenden Ergebnissen. Eine geringere Leistung entspricht kleineren Werten der Verstärkungsfaktoren. Da kleinere Reglerverstärkungen zu einem langsameren Reaktionsverhalten der Zylinder führen, entspricht dies den Erwartungen. Die Verstärkungen sind nach oben begrenzt und nehmen nicht die maximal möglichen Werte an, da das System bei zu großer Verstärkung instabil wird. Für instabile Systeme nehmen sowohl der Energieverbrauch als auch die Abweichungen der Ist- von der Sollkraft zu. Derartige Konfigurationen können daher nicht paretooptimal sein.

Bemerkenswert an der Paretomenge ist, dass die Parameterwerte nicht gleichmäßig ansteigen, d. h. auf einer Geraden liegen. Hier kommt die bauliche Anordnung der Hydraulikzylinder und der daraus resultierende variierende Einfluss auf die beiden Zielgrößen zur Geltung. Für kleine Verstärkungen ist deutlich zu erkennen, dass vor allem durch eine Erhöhung von $K_{p,3}$ die Regelgüte verbessert wird. Eine Vergrößerung der anderen Faktoren trägt ohne Zweifel auch zur Verbesserung der Regelgüte bei. Allerdings wird hierfür – so die Interpretation des Optimierungsergebnisses – deutlich mehr Energie für die gleiche Erhöhung der Regelgüte benötigt. Mit anderen Worten wäre eine derartige Konfiguration nicht paretooptimal. Dies verdeutlicht einen der Vorteile der Mehrzieloptimierung. Derartige Einflüsse lassen sich durch eine manuelle Analyse des Systemverhaltens nicht oder nur mit sehr großem Aufwand erkennen.

4.4.2 Mehrzieloptimierung des Feder-Neige-Prüfstands

Als zweites Anwendungsbeispiel einer einfachen, nicht hierarchischen Mehrzieloptimierung wird die Regelung des gesamten Feder-Neige-Prüfstands optimiert. Modelle der beiden Aktormodule aus dem vorhergehenden Abschnitt werden in linearer Form berücksichtigt. Im Gegensatz zu dem hierarchischen Optimierungsansatz, der in Abschnitt 6.1 behandelt wird, wird innerhalb dieses Abschnitts eine feste Konfiguration für beide Aktormodule verwendet. Dazu wird ein fester Punkt aus der Paretomenge der Aktormodule mit moderatem dynamischem Verhalten gewählt.

Die gesamte Aufbauregelung des Feder-Neige-Prüfstands besteht aus einer relativen Federung und Dämpfung sowie einem Sky-Hook Anteil, siehe [VT08] für weitere Details zur Regelungsstrategie. Zur Vereinfachung wird im Folgenden nur die Sky-Hook Regelung betrachtet. Dabei werden aus den gemessenen Beschleunigungen der Aufbaumasse die von den Aktormodulen zu stellenden Kräfte berechnet. Die Beschleunigungen werden mit Hilfe geeigneter Hochpass- und Tiefpassfilter modifiziert, sodass für die Regelung Approximationen der absoluten Geschwindigkeiten der drei Freiheitsgrade des Aufbaus zur Verfügung stehen. Proportional zu diesen Geschwindigkeiten wird die zu stellende Kraft über drei Faktoren d_z, d_y und d_φ berechnet und mittels kinematischer Umformungen in Sollkräfte der Aktormodule umgerechnet.

Für die Mehrzieloptimierung werden erneut zwei Zielgrößen definiert. Die erste Zielfunktion stellt den Regelungsaufwand als Maß für den Energieverbrauch dar. Er kann durch die zu realisierenden Sollkräfte angenähert werden, ähnlich wie dies auch im Rahmen der Linear-Quadratischen-Regelungen geschieht.

Die zweite Zielfunktion beschreibt den durch die aktive Federung erzielten Komfort. Diese Zielgröße setzt sich zusammen aus den Aufbaubeschleunigungen, die im realen Fahrzeug auf die Insassen wirken würden. Untersuchungen zur Auswirkungen von Schwingungen auf das menschliche Wohlbefinden folgend, werden die Beschleunigungen frequenzabhängig gewichtet. Hierfür werden die in [VDI04a] beschriebenen Bandpassfilter eingesetzt. Der Komfort kann dann durch den zeitlichen Mittelwert der gewichteten Aufbaubeschleunigungen berechnet werden.

Als Zielfunktionen werden sowohl für den Energieverbrauch als auch für den Komfort quadratische, zeitliche Mittelwerte

$$f_{1,2} : \mathbb{R}^3 \rightarrow \mathbb{R}, p \mapsto f_{1,2}(p) = \frac{1}{T} \int_0^T y(t)^T Q_{1,2} y(t) dt \quad (4-5)$$

verwendet. Der Vektor y beschreibt hierbei den Ausgang aus dem Bewertungsmodell, der die Sollkräfte und die gewichteten Beschleunigungen enthält. Die

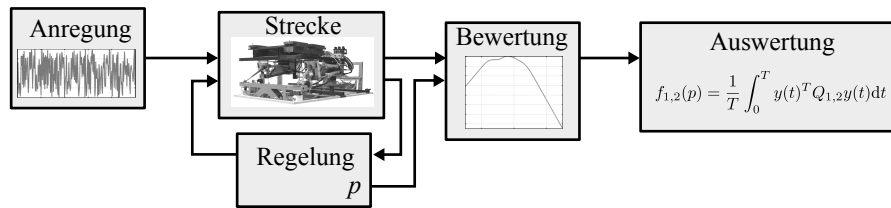


Bild 4-7: Optimierungsmodell des gesamten Feder-Neige-Prüfstands

Matrizen Q_1 und Q_2 stellen Diagonalmatrizen dar und dienen neben der Auswahl der jeweiligen Komponenten aus y noch der Skalierung der Zielfunktionen. Die Skalierung basiert auf Untersuchungen am Optimierungsmodell und ist so gewählt, dass die Zielgrößen in einem annähernd gleichen Wertebereich liegen.

Das Optimierungsmodell ist in Bild 4-7 dargestellt und wird nachfolgend kurz erläutert. Das Bewertungsmodell enthält, wie bereits erwähnt, die Frequenzgewichtung der Aufbaubeschleunigungen.

Als Streckenmodell wird ein lineares Modell des Feder-Neige-Prüfstands verwendet. Die Aktormodule werden zuvor separat linearisiert und dann in linearisierter Form in das Gesamtmodell eingebunden. Als Anregung wird erneut ein bandbegrenztes, Gauß'sches weißes Rauschen verwendet, allerdings nicht so hochfrequent wie bei der Optimierung des Aktormoduls. Aufgrund der größeren Trägheiten im Streckenmodell hätte eine höherfrequente Anregung kaum Auswirkungen auf das Ergebnis. Das Anregungssignal nähert die Störungen des Schienenverlaufs an. Es besteht im Optimierungsmodell aus den Sollvorgaben, die von der Anregungseinheit umgesetzt werden. Da die Störungen keinen direkten Einfluss auf die Zielfunktionen haben, besteht keine direkte Verbindung der Anregung mit der Bewertung, anders als im Optimierungsmodell des Aktormoduls. Die Regelung besteht aus der Signalverarbeitung der Messgrößen sowie der Sky-Hook Regelung.

Analog zur Mehrzieloptimierung des Aktormoduls wird eine Simulation mit fest vorgegebener Simulationszeit T durchgeführt. Das Anregungssignal ist in jeder Simulation gleich. Lediglich die Optimierungsparameter, d. h. die Sky-Hook Parameter $p = [d_z, d_y, d_\varphi]$, werden variiert. Das resultierende Mehrzieloptimierungsproblem

$$\min_p \{ [f_1(p), f_2(p)] : p \in \mathcal{D} \subseteq \mathbb{R}^3 \}.$$

wird erneut mit den Unterteilungsalgorithmen aus Abschnitt 4.1 gelöst. Die Reglerparameter beschreiben die Dämpfung der Aufbaumasse und sind daher stets positiv. Zusätzlich werden obere Schranken für die Parameter festgelegt, für die

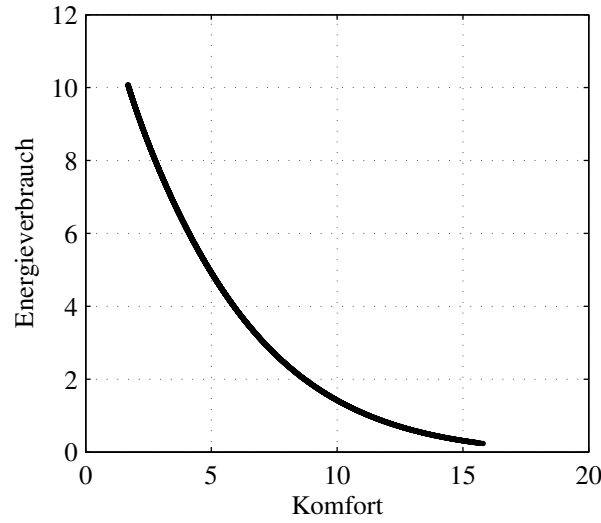


Bild 4-8: Paretofront des Feder-Neige-Prüfstands

Hubbewegung auf $28 \frac{kNs}{m}$, für die Querbewegung auf $6 \frac{kNs}{m}$ und für die Wankbewegung auf $6 \frac{kNs}{rad}$.

Die resultierende Paretofront ist in Bild 4-8 zu finden. Sie weist wieder die gleiche typische Form auf, die auch bei der Optimierung des Aktormoduls zu beobachten war. Auch hier wird deutlich, dass die beiden Zielgrößen gegenläufig sind und kein gemeinsames Minimum besitzen, was erneut mit der physikalischen Anschauung übereinstimmt.

Die Paretomenge im Urbildraum und die zugehörige Boxüberdeckung nach 23 Unterteilungsschritten sind in Bild 4-9 dargestellt. Trotz der relativ hohen Streuung der berechneten Paretopunkte ist gut zu erkennen, dass es sich um eine zusammenhängende, eindimensionale Paretomenge handelt. Sie erstreckt sich über den gesamten Definitionsbereich von d_z und d_ϕ , die Veränderungen des verbleibenden Parameters d_y sind deutlich kleiner. Dies kann in der Weise interpretiert werden, dass der Komfort in y-Richtung schlechter beeinflusst werden kann.

Die Paretomenge beginnt eigentlich im Ursprung des Parameterraumes. Dies ist zu erwarten gewesen, da im Nullpunkt die Sollkräfte die gesamte Simulationszeit über verschwinden und somit die Energie-Zielfunktion ihr Minimum annimmt. Allerdings ändert sich an diesem Punkt die Systemdynamik grundlegend, da im Nullpunkt ein passives System vorliegt. Dies führt bei der anschließenden Modellordnungsreduktion zu Problemen, die auf diese Strukturänderung des Modells zurückzuführen sind und eine weitergehende Untersuchung erfordern, die den Rahmen der vorliegenden Arbeit übersteigen würde, siehe [KT12] für einige Beobachtungen zu dieser Thematik. Da die hier vorgestellten Optimierungsergeb-

nisse in den nachfolgenden Kapiteln aufgegriffen werden, wird die Paretomenge nachträglich ein wenig verkürzt.

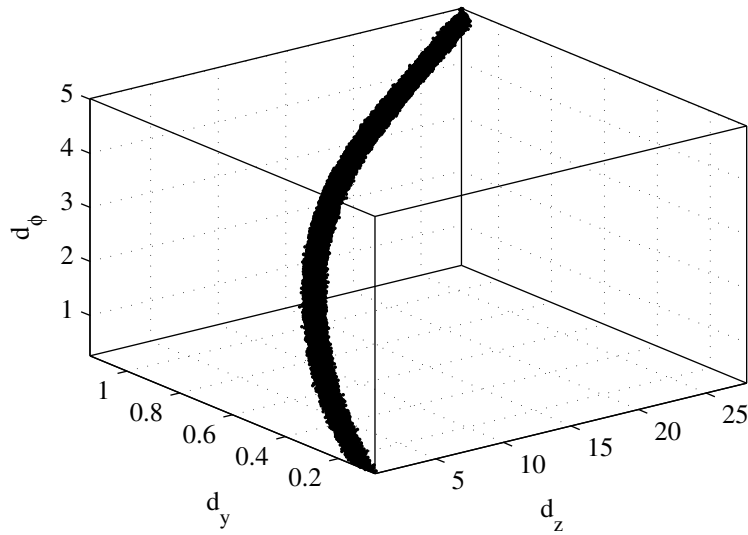


Bild 4-9: Paretomenge des Feder-Neige-Prüfstands

5 Parametrische Modellordnungsreduktion paretooptimaler Systeme

Die im vorhergehenden Kapitel vorgestellte Mehrzieloptimierung eignet sich dazu, optimale Systemkonfigurationen für selbstoptimierende Systeme zu berechnen. Die zuvor in Kapitel 3 beschriebenen Verfahren der parametrischen Modellordnungsreduktion (PMOR) dienen dazu, lineare dynamische Systeme zu vereinfachen und die Abhängigkeit von bestimmten Parametern auch im reduzierten System zu erhalten. In diesem Kapitel wird ein neuartiger Weg vorgestellt, beide Methoden, die Mehrzieloptimierung und die PMOR, miteinander zu verknüpfen.

Mit der im Folgenden dargestellten Vorgehensweise ist es möglich, reduzierte Systeme zu erzeugen, die direkt von Zielvorgaben abhängen. Dies ist besonders im Kontext hierarchischer selbstoptimierender Systeme von Vorteil. Unterlagerte Teilsysteme können so einerseits reduziert werden, was die Komplexität des Systems verringert, bleiben aber andererseits in gewissem Maß variabel bzgl. Parameteränderungen. Dies ist beispielsweise für eine Optimierung des Gesamtsystems vorteilhaft. Die Kombination von Mehrzieloptimierung mit PMOR ermöglicht die Verwendung abstrakter Zielvorgaben an Stelle von physikalischen Parametern. Im Hinblick auf die multidisziplinäre Entwicklung selbstoptimierender Systeme ist dieser Aspekt vorteilhaft. Das reduzierte Teilsystem kann in gekapselter Form an Dritte weitergegeben werden. Für die Parametervariation selbst ist nur wenig Wissen über das System erforderlich. Die Verhaltensanpassung kann durch abstrakte Zielvorgaben, beispielsweise den Energieverbrauch zu verringern, umgesetzt werden.

Die in dieser Arbeit entstandene Methode besteht aus drei wesentlichen Schritten. Als Ausgangspunkt dient die Paretomenge, d. h. die durch ein Optimierungsverfahren berechneten optimalen Kompromisse mit den zugehörigen Systemkonfigurationen. Als erstes ist eine geeignete Parametrisierung dieser Paretomenge zu definieren, mit der Zielvorgaben in Systemkonfigurationen umgerechnet werden können. Dies ist in Abschnitt 5.1 ausführlich dargestellt. Der zweite Schritt, beschrieben in Abschnitt 5.2, besteht aus einer Aufbereitung der zu den optimalen Konfigurationen gehörenden Systeme mit Hilfe von Interpolationsverfahren. Hierdurch wird eine effiziente PMOR ermöglicht. Als letzter Schritt werden in Abschnitt 5.3 Verfahren der PMOR auf die interpolierten Systeme angewendet.

Zwei Voraussetzungen müssen für die Anwendbarkeit dieser Methode erfüllt sein. Zum einen kann die PMOR, wie sie in dieser Arbeit betrachtet wird, nur auf li-

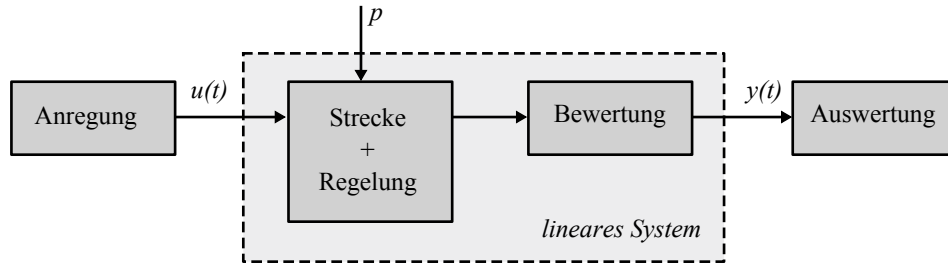


Bild 5-1: Lineares Optimierungsmodell für die simulationsbasierte Mehrzieloptimierung selbstoptimisierender Systeme

neare Systeme angewendet werden. Je nach Einsatzgebiet des reduzierten Modells wird es sich hierbei um ein Modell der geregelten Strecke mit gewissen physikalischen Ein- und Ausgängen handeln. Dies ist insbesondere der Fall, wenn das reduzierte System ein physikalisch gekoppeltes Teilsystem darstellt. Das Aktor-Modul des Feder-Neige-Prüfstands, betrachtet in Abschnitt 5.4.1, ist ein Beispiel hierfür.

Auf höheren Hierarchieebenen können die Ein- und Ausgänge des zu reduzierenden Systems aber auch denen des Optimierungsmodells aus Abschnitt 4.3 entsprechen. Dann werden mit dem reduzierten System direkt die Zielgrößen berechnet. Der Feder-Neige-Prüfstand im Kontext des vernetzten Prüfstands aus Abschnitt 6.2 ist ein Beispiel hierfür. Numerische Ergebnisse der Reduktion sind in Abschnitt 5.4.2 zu finden. Auf die weitere Verwendung der parametrischen reduzierten Systeme, sowohl für den Feder-Neige-Prüfstand, als auch für das Aktor-Modul, wird ausführlich in Kapitel 6 eingegangen.

Für die weiteren Betrachtungen werden, falls erforderlich, alle dynamischen Anteile des Optimierungsmodells wie in Bild 5-1 dargestellt zu einem linearen System zusammengefasst. Da die Optimierungsparameter der Regelung entstammen, wird weiterhin angenommen, dass die Eingangs- und Ausgangsmatrix nicht von den Parametern abhängen. Man erhält dann das lineare, parameterabhängige System

$$\begin{aligned}\dot{x}(t) &= A(p)x(t) + Bu(t), \\ y(t) &= Cx(t).\end{aligned}\tag{5-1}$$

Die Art der Parameterabhängigkeit ist hierbei beliebig, solange die Systemmatrix stetig bzgl. der Parameter ist. Ein Durchgriff vom Systemeingang u auf den Systemausgang y wird vernachlässigt. Dieser spielt für die Modellordnungsreduktion keine Rolle und kann, falls er in einem konkreten System vorhanden ist, unverändert ins reduzierte System übernommen werden.

Als zweite wesentliche Voraussetzung werden nur Mehrzieloptimierungsprobleme mit zwei Zielfunktionen betrachtet, die über eine eindimensionale, zusammenhän-

gende Paretomenge verfügen. Für die in dieser Arbeit betrachteten Anwendungsbeispiele sind diese Voraussetzungen erfüllt. Auf mögliche Erweiterungen der in diesem Kapitel vorgestellten Methode wird in Abschnitt 5.5 eingegangen.

5.1 Parametrisierung von Paretomengen

Bei Mehrzieloptimierungsproblemen mit zwei Zielfunktionen stellt die Paretomenge unter gewissen Voraussetzungen an die Zielfunktionen eine eindimensionale Mannigfaltigkeit dar, siehe hierzu auch Kapitel 4. Dies bedeutet unter anderem, dass die Paretomenge durch einen skalaren Parameter parametrisiert werden kann. Dieser Abschnitt beschäftigt sich mit der Erstellung derartiger Parametrisierungen.

Als Ausgangspunkt wird angenommen, dass zu einem gegebenen Mehrzieloptimierungsproblem (4-1) bereits eine Approximation P_F der Paretomenge vorliegt, berechnet beispielsweise durch die Anwendung der mengenorientierten Verfahren, die in Abschnitt 4.1 beschrieben sind. Die numerische Approximation der Paretomenge besteht aus einer endlichen Anzahl N_{P_F} nichtdominierter Punkte, d. h.

$$P_F = \{p_j^* \in \mathbb{R}^{n_p}, 1 \leq j \leq N_{P_F}\}.$$

Das Ziel besteht nun darin, eine Funktion

$$s : [0, \alpha_{max}] \rightarrow \mathbb{R}^{n_p}, \alpha \mapsto s(\alpha)$$

zu definieren, die nur von dem skalaren Parameter α abhängt und die Paretomenge P_F möglichst gut annähert.

Über die Lage der Paretomenge im Urbild- bzw. Parameterraum kann keine allgemeine Aussage getroffen werden. Wie bereits erwähnt, wird an dieser Stelle lediglich angenommen, dass es sich um eine zusammenhängende Menge handelt. Bei stetigen, gegenläufigen Zielfunktionen liegt dann im Bildraum die bereits mehrfach dargestellte hyperbelartige Form vor. Zudem besteht für die Anwendung eine Anforderung darin, Zielvorgaben mit Hilfe des Parameters α auszudrücken. Diese Zielvorgaben beziehen sich ebenfalls auf den Bildraum. Beides spricht dafür, die Paretofront, d. h. das Bild $F(P_F)$ als Ausgangspunkt für die Parametrisierung zu verwenden.

Eine wichtige Rolle spielen im Folgenden die Endpunkte der Paretofront, die von den Minima der einzelnen Zielfunktionen

$$\begin{aligned} p^I &= \arg \min_{p \in \mathcal{D}} f_1(p), \\ p^{II} &= \arg \min_{p \in \mathcal{D}} f_2(p), \end{aligned}$$

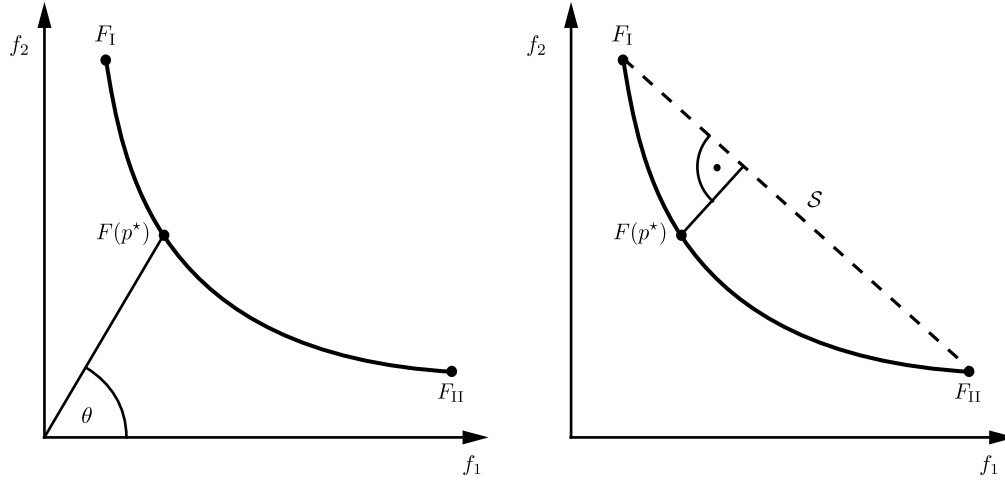


Bild 5-2: Parametrisierungsmöglichkeiten der Paretofront; links: Winkel θ des Zielfunktionsvektors; rechts: orthogonale Projektion auf den Simplex \mathcal{S}

mit den zugehörigen Punkten im Bildraum

$$F_I := F(p^I) = [f_1(p^I) \quad f_2(p^I)]^T,$$

$$F_{II} := F(p^{II}) = [f_1(p^{II}) \quad f_2(p^{II})]^T,$$

gebildet werden.

Es sind viele Ansätze zur Parametrisierung denkbar, von denen in dieser Arbeit zwei verschiedene verwendet werden. Zum einen wird der Winkel der Zielfunktionsvektoren zur Beschreibung genutzt. Zum anderen werden die Paretopunkte mit Hilfe eines Simplex zwischen den beiden Minima F_I und F_{II} parametrisiert. Beide Varianten sind in Bild 5-2 dargestellt und werden im Folgenden detailliert beschrieben.

Der Winkel θ eines Zielfunktionsvektors $F(p^*)$ zu einem gegebenen Paretopunkt $p^* \in P_F$ ist gegeben als

$$\theta(p^*) = \text{atan} \left(\frac{f_2(p^*)}{f_1(p^*)} \right).$$

Alle Punkte der Paretofront können auf diese Weise eindeutig beschrieben werden. Betrachtet man eine äquidistante Diskretisierung der Winkel, so führt dies zu einer dichteren Punkteverteilung in der Mitte der Paretofront und größeren Abständen in den Randbereichen für den Fall, dass die Paretofront eine relativ hohe Krümmung aufweist. Dies kann bei der weiteren Verwendung der Paretofront ein Nachteil sein, wenn eine gleichmäßige Diskretisierung im Bildraum gewünscht

ist. Beispielsweise ist dies bei der hybriden Planung auf Basis von Paretomengen, siehe [MAK⁺08], der Fall. Für die hier vorgestellte Methode der parametrischen Modellordnungsreduktion paretooptimaler Systeme ist dies jedoch nicht von Bedeutung. Vorteilhaft bei der Verwendung des Winkels θ ist die leichtere geometrische Interpretation der Parametrisierung, die u. a. bei der Umsetzung der zielfunktionsbasierten Regelung in Abschnitt 6.3 zum Tragen kommt.

Für die weitere Verwendung wird der relative Winkel bezogen auf die Endpunkte der Paretofront betrachtet. Zusätzlich kann die Parametrisierung über den Faktor α_{max} auf ein gewünschtes Intervall $[0, \alpha_{max}]$ skaliert werden, woraus sich zu jedem Paretopunkt p^* als Parametrisierung

$$\alpha(p^*) = \alpha_{max} \cdot \frac{\theta(p^*) - \theta(p^I)}{\theta(p^I) - \theta(p^II)} \quad (5-2)$$

ergibt.

Die zweite Variante zur Parametrisierung verwendet einen Simplex $\mathcal{S} = F_I - F_{II}$, der von den beiden Randpunkten F_I und F_{II} der Paretofront gebildet wird. Eine Parametrisierung aller Paretopunkte kann, in Anlehnung an die NBI-Methode zur Mehrzieloptimierung aus Kapitel 4, durch eine orthogonale Projektion auf den Simplex erfolgen. Dies führt tendenziell zu einer gleichmäßigeren Punkteverteilung bei einer äquidistanten Diskretisierung bzgl. der Parametrierung als bei der Verwendung des Winkels der Paretopunkte. Eine direkter Bezug zur Gewichtung der Zielfunktionen geht jedoch verloren.

Zu einem gegebenen Paretopunkt p^* berechnet sich der zugehörige Parameterwert als

$$\alpha(p^*) = \alpha_{max} \cdot \frac{\mathcal{S}^T (F(p^*) - F_{II})}{\|\mathcal{S}\|_2^2}. \quad (5-3)$$

Auch diese Parametrierung kann durch Anpassung des Faktors α_{max} auf ein gewünschtes Intervall $[0, \alpha_{max}]$ skaliert werden.

Statt einer orthogonalen Projektion auf den Simplex kann alternativ ein zusätzlicher Punkt Q in Kombination mit dem Simplex verwendet werden. Als Parametrisierung α eines Paretopunktes p^* wird dann der Schnittpunkt der Geraden durch die beiden Punkte Q und $F(p^*)$ mit dem Simplex \mathcal{S} verwendet. Diese Art der Parametrierung wurde in [Mün12] für die Mehrzieloptimierung sowie die zielfunktionsbasierte Regelung verwendet. Mit ihrer Hilfe kann die Verteilung der Punkte bei äquidistanten α -Werten zusätzlich beeinflusst werden. Insbesondere dichter werdende Punkte in den Randgebieten der Paretofront sind auf diese Weise möglich, wenn der Punkt Q oberhalb der Paretofront, z. B. durch $Q = [F_{II} \ F_I]$, definiert wird. Da von dieser speziellen Parametrisierungsvariante jedoch kein

Mehrwert für die in dieser Arbeit beschriebene Methodik zu erwarten ist, wird sie nicht verwendet.

Bei Verwendung des mengenorientierten Ansatzes zur Berechnung der Paretofront, wie er im vorhergehenden Kapitel 4 beschrieben ist, erhält man als Ergebnis eine Approximation der Paretomenge, bei der die berechneten Paretopunkte eine relativ hohe Streuung aufweisen. Die dargestellten Optimierungsergebnisse in Abschnitt 4.4 verdeutlichen dies. Nach einer ersten Parametrisierung, d. h. der Berechnung von $\tilde{\alpha}_j$, $1 \leq j \leq N_{PF}$ für alle Paretopunkte p_j^* gemäß (5-2) oder (5-3), wird daher eine Ausgleichskurve berechnet. Diese ermöglicht eine glatte eindimensionale Approximation der Paretomenge.

Hierfür wird ein sogenannter *smoothing spline* $g : \mathbb{R} \rightarrow \mathbb{R}^{n_p}$, $\tilde{\alpha} \mapsto g(\tilde{\alpha})$ benutzt. Dieser stellt einen kubischen Spline dar, der das Minimierungsproblem

$$\min_g \left\{ \gamma \sum_{j=1}^{N_{PF}} w_j \|p_j^* - g(\alpha_j)\|^2 + (1 - \gamma) \int \lambda(\alpha) \left\| \frac{\partial^2 g}{\partial \alpha^2} \right\|^2 d\alpha \right\} \quad (5-4)$$

löst. Das Minimierungsproblem besteht aus zwei wesentlichen Bestandteilen. Einerseits wird die quadratische Abweichung der Paretopunkte von den Werten des Splines, andererseits die Krümmung des Splines minimiert. Durch eine manuelle Variation der Gewichtung γ kann eine glatte Ausgleichskurve durch die gestreuten Paretopunkte erzeugt werden. Die Berechnung des *smoothing spline* erfolgte durch die *Curve Fitting Toolbox* in Matlab [Mat11]. Für die beiden zusätzlichen Parameter w_j und λ wurden die von Matlab vordefinierten Standardwerte benutzt, da mit ihnen ausreichend gute Ergebnisse erzielt werden konnten.

Das Resultat ist ein kubischer Spline $g(\tilde{\alpha})$ mit den Stützstellen $\tilde{\alpha}_j$, der die Paretomenge approximiert. Allerdings ist die ursprüngliche Bedeutung von $\tilde{\alpha}$ durch die Berechnung des Ausgleichssplines verloren gegangen. Daher wird als letzter Schritt die Parametrisierung auf Basis der Punkte $F(g(\tilde{\alpha}_j))$ erneut durchgeführt, wodurch die finalen Parameterwerte α_j entstehen. Die Parametrisierungsfunktion s , die für das weitere Vorgehen verwendet wird, ist dann erneut ein kubischer Spline, basierend auf den Paaren $[\alpha_j, g(\tilde{\alpha}_j)]$. Die ursprüngliche Approximation der Paretomenge besteht oftmals aus einer sehr hohen Zahl nichtdominierter Punkte. Für den endgültigen kubischen Spline können weniger Stützstellen verwendet werden. Eine Vorgabe der gewünschten Anzahl ist in der Standardfunktionalität der *Curve Fitting Toolbox* enthalten, die im Rahmen dieser Arbeit hierfür verwendet wurde.

Zusammengefasst sind zur Parametrisierung einer Paretomenge die folgenden vier Schritte auszuführen.

- (i) Parametrisierung der nichtdominierten Punkte $p_j^* \in P_F$ mit Hilfe von (5-2) oder (5-3)

- (ii) Berechnung eines *smoothing spline* $g(\tilde{\alpha}_j)$ zur Glättung der nichtdominierten Punkte
- (iii) erneute Parametrisierung der geglätteten Paretopunkte $F(g(\tilde{\alpha}_j))$
- (iv) Berechnung eines kubischen Splines $s : \mathbb{R} \rightarrow \mathbb{R}^{n_p}$, $\alpha \mapsto s(\alpha)$ mit $s(\alpha_j) = g(\tilde{\alpha}_j)$ und evtl. Verringerung der Anzahl der Stützstellen

5.2 Interpolation paretooptimaler Systeme

Mit Hilfe der im vorhergehenden Abschnitt beschriebenen Parametrisierung der Paretomenge kann diese durch einen skalaren Parameter α beschrieben werden. Setzt man diese Parametrisierung in die Systembeschreibung (5-1) ein, so ergibt sich

$$\begin{aligned} \dot{x}(t) &= A(s(\alpha))x(t) + Bu(t), \\ y(t) &= Cx(t), \quad \alpha \in [0, \alpha_{max}]. \end{aligned} \tag{5-5}$$

Der Vorteil liegt darin, dass auch die dynamischen Systeme, die zu den paretooptimalen Konfigurationen gehören, mit Hilfe des skalaren Parameters α ausgedrückt werden können. Diese linearen, parametrischen Systeme werden dementsprechend als *paretooptimale Systeme* bezeichnet. Das Ziel der in diesem Kapitel beschriebenen Methode besteht darin, die paretooptimalen Systeme mit Hilfe der parametrischen Modellordnungsreduktion zu reduzieren. Hierfür ist es für alle im Rahmen dieser Arbeit betrachteten Verfahren von Vorteil, wenn die Systeme von wenigen Parametern abhängen. Bei der manuellen rationalen Interpolation, siehe Abschnitt 3.1, erhöht sich tendenziell die Ordnung der reduzierten Systeme, bei der Matrix Interpolation aus Abschnitt 3.2 steigt die Komplexität des Verfahrens, da mehr Diskretisierungspunkte im Parameterraum benötigt werden und insbesondere die Bestimmung geeigneter Gewichtsfunktionen zur Interpolation der Systeme deutlich aufwändiger ist. Da die Parametrisierung $s(\alpha)$ nur von der Anzahl der Zielfunktionen, nicht jedoch von der Anzahl der Optimierungsparameter p abhängt, führt eine Verwendung von α als Reduktionsparameter besonders für Systeme mit einer hohen Anzahl n_p von Optimierungsparametern zu einer erheblichen Vereinfachung.

Beide Reduktionsverfahren können jedoch nicht direkt auf die paretooptimalen Systeme (5-5) angewendet werden. Für die manuelle rationale Interpolation ist eine affin-lineare Parameterabhängigkeit notwendig, die für $A(s(\alpha))$ nicht vorausgesetzt werden kann. Die Matrix Interpolation erfordert die Definition geeigneter Diskretisierungspunkte im Parameterraum sowie passende Gewichtsfunktionen für die Interpolation der Matrizen. Der im Folgenden beschriebene Ansatz zur Interpolation paretooptimaler Systeme ist für beide Varianten gleichermaßen geeignet und stellt sicher, dass alle Voraussetzungen erfüllt sind.

Basierend auf der Parametrisierung α wird eine Diskretisierung bestehend aus einer endlichen Zahl n_S von Stützstellen

$$0 = \alpha_1 < \alpha_2 < \dots < \alpha_{n_S} = \alpha_{max}$$

definiert. Mit Hilfe dieser Diskretisierung wird eine weitere Interpolation vorgenommen, allerdings nicht der Paretopunkte $s(\alpha)$ im Parameterraum, sondern der paretooptimalen Systeme (5-5). Da die Ein- und Ausgangsmatrizen nicht von dem Parameter α abhängen, wird zunächst nur die Systemmatrix $A(s(\alpha))$ betrachtet. Unter Verwendung der Stützstellen α_i lässt sich eine lineare Interpolation $mpp : \mathbb{R} \rightarrow \mathbb{R}^{n_x \times n_x}$, $\alpha \mapsto mpp(\alpha)$ mit

$$mpp(\alpha) = \underbrace{A(s(\alpha_i))}_{A_i} + \underbrace{\frac{\alpha - \alpha_i}{\alpha_{i+1} - \alpha_i}}_{\bar{\alpha}_i} \underbrace{[A(s(\alpha_{i+1})) - A(s(\alpha_i))]}_{\Delta A_i}, \quad (5-6)$$

für $\alpha \in [\alpha_i, \alpha_{i+1})$ erstellen. Diese Interpolation ist stetig bzgl. α , was eine grundlegende Anforderung für die weitere Verwendung der reduzierten Systeme beispielsweise im Rahmen der hierarchischen Optimierung darstellt. Darüber hinaus ist die Interpolation eindeutig durch die Systemmatrizen an den Stützstellen A_i definiert. Dies ist insbesondere bei komplexen Systemen von Vorteil, bei denen die Berechnung der Systemmatrix zu gegebenen Parametern oftmals aufwändig ist. Analog zur linearen Interpolation der Systemmatrizen sind auch Interpolationen höherer Ordnung denkbar. Hierauf wird in Abschnitt 5.5 eingegangen.

Die Voraussetzungen an die Reduktionsverfahren sind durch die lineare Interpolation erfüllt. Für jedes Intervall liegt nun eine affin-lineare Parameterabhängigkeit

$$A(\alpha) = A_i + \bar{\alpha}_i \Delta A_i$$

vor, sodass die manuelle rationale Interpolation intervallweise angewendet werden kann.

Als Stützstellen für die Matrix Interpolation werden die Systemmatrizen an den Diskretisierungspunkten A_i verwendet. Betrachtet man jedes Intervall separat, so ergibt sich durch Umstellen von (5-6) eine Interpolation von A_i und A_{i+1} in der für die Matrix Interpolation geforderten Form:

$$A(\alpha) = (1 - \bar{\alpha}_i) A_i + \bar{\alpha}_i A_{i+1}.$$

Erweitert auf die gesamten paretooptimalen Systeme sind die Gewichtsfunktionen ω_i , die in (3-9) für die parametrischen reduzierten Systeme benötigt werden, gegeben als

$$\omega_i(\alpha) = \begin{cases} \frac{\alpha - \alpha_{i-1}}{\alpha_i - \alpha_{i-1}} & \text{für } \alpha \in [\alpha_{i-1}, \alpha_i) \\ \frac{\alpha_{i+1} - \alpha}{\alpha_{i+1} - \alpha_i} & \text{für } \alpha \in [\alpha_i, \alpha_{i+1}) \\ 0 & \text{sonst} \end{cases} \quad (5-7)$$

Die Wahl der Diskretisierungspunkte α_i hat einen erheblichen Einfluss auf das Reduktionsergebnis. Hierbei stehen zwei gegensätzliche Ziele einander gegenüber. Zum einen ist es vorteilhaft, möglichst wenige Diskretisierungspunkte zu benutzen, da der Aufwand zur Berechnung des parametrischen reduzierten Systems bei beiden Reduktionsverfahren mit der Anzahl der Diskretisierungspunkte steigt. Zum anderen soll eine möglichst hohe Approximationsgüte der reduzierten Systeme verglichen mit den ursprünglichen paretooptimalen Systemen erreicht werden. Diese nimmt mit wachsender Stützstellenzahl zu. Im Rahmen dieser Arbeit ist daher ein Algorithmus entstanden, der die Diskretisierungspunkte automatisiert und effizient bestimmt und dabei versucht, mit möglichst wenig Stützstellen eine hohe Approximationsgüte zu erzielen.

Der Algorithmus besteht aus zwei wesentlichen Teilen, einer Neuverteilung der bestehenden Stützstellen und dem Hinzufügen weiterer Stützstellen. Die Neuverteilung der Stützstellen basiert auf dem *newnot*-Algorithmus, der zu den etablierten Verfahren aus dem Bereich der Spline-Interpolation zählt. Die grundsätzliche Funktionsweise ist nachfolgend kurz für die lineare Spline-Interpolation umrissen. Eine ausführliche Herleitung für den allgemeinen Fall ist in [Boo01] zu finden.

Den Ausgangspunkt bildet die allgemeine Fehlerabschätzung eines linearen Splines l bezogen auf ein bestimmtes Intervall $[\alpha_i, \alpha_{i+1}]$

$$\|g - l\|_{[\alpha_i, \alpha_{i+1}]} \leq C \cdot (\alpha_{i+1} - \alpha_i)^2 \|D^2 g\|_{[\alpha_i, \alpha_{i+1}]},$$

wobei $g : \mathbb{R} \rightarrow \mathbb{R}$ die zu interpolierende Funktion, $C \in \mathbb{R}$ eine positive Konstante und D den Differentialoperator darstellen. Die hier verwendete Norm $\|\cdot\|_{[\alpha_i, \alpha_{i+1}]}$ bezeichnet die Maximumnorm $\|\cdot\|_\infty$ einer Funktion eingeschränkt auf das angegebene Intervall $[\alpha_i, \alpha_{i+1}]$. Aus der Fehlerabschätzung geht hervor, dass die Approximationsgüte des linearen Splines l abnimmt, wenn die zu interpolierende Funktion eine große Krümmung aufweist. Da auch die Intervalllänge in die Abschätzung eingeht, ist ein Ansatz zur Verbesserung der Interpolation, die Stützstellen so zu wählen, dass kleinere Intervalle in Bereichen mit großer Krümmung vorliegen.

Betrachtet man nun alle Intervalle gleichzeitig, so ergibt sich unter zusätzlichen Vereinfachungen für eine optimale Stützstellenverteilung das Ziel, die Funktion

$$\max_i \|D^2 g\|_{[\alpha_i, \alpha_{i+1}]} (\alpha_{i+1} - \alpha_i)^2$$

zu minimieren. Das Minimum dieser Funktion wird dann erreicht, wenn die Intervallgrenzen so gewählt sind, dass $\|D^2 g\|_{[\alpha_i, \alpha_{i+1}]} (\alpha_{i+1} - \alpha_i)^2$ konstant ist für alle Intervalle. Äquivalent hierzu müssen die Stützstellen so gewählt werden, dass

$$\int_{\alpha_i}^{\alpha_{i+1}} |D^2 g(\tau)|^{1/2} d\tau = \frac{1}{n_S - 1} \int_0^{\alpha_{max}} |D^2 g(\tau)|^{1/2} d\tau$$

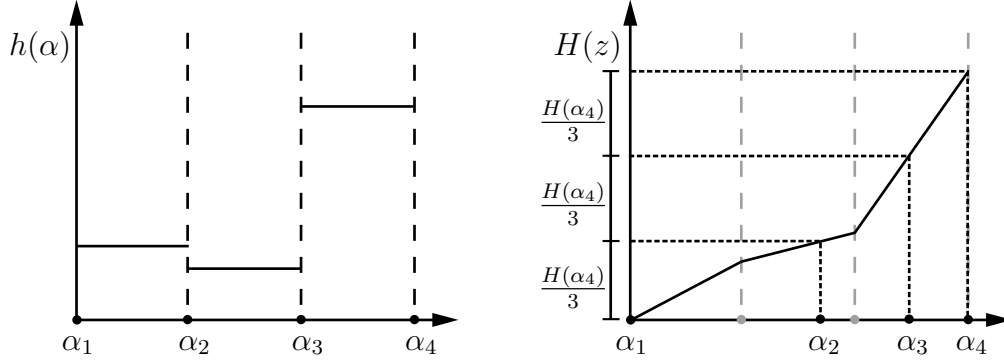


Bild 5-3: Prinzipielle Funktionsweise des newnot-Algorithmus links: stückweise konstante Approximation der Krümmung basierend auf den bisherigen Stützstellen rechts: Bestimmung der verbesserten Stützstellen durch eine äquidistante Aufteilung des Integrals H

für $i = 1, \dots, n_S - 1$ erfüllt ist.

Um eine explizite Berechnung der Krümmung $|D^2g|$ von g zu vermeiden, nutzt der newnot-Algorithmus eine stückweise konstante Approximation

$$h \approx |D^2g|,$$

die ausschließlich mit Hilfe des linearen Splines l berechnet werden kann. Die genaue Definition der Approximation h ist in [Boo01] nachzulesen. Ist diese Funktion bekannt, lässt sich die stetige und monoton wachsende Funktion

$$H(z) = \int_0^z (h(\tau))^{1/2} d\tau \quad (5-8)$$

definieren. Da G monoton wachsend ist, kann die verbesserte Verteilung der Stützstellen mittels der Umkehrfunktion H^{-1} berechnet werden. Hierzu werden die $n_S - 2$ äquidistanten Punkte $i \cdot H(\alpha_{max})/(n_S - 1)$ verwendet, d. h.

$$\alpha_{i+1} = H^{-1}(i \cdot H(\alpha_{max})/(n_S - 1)), \quad i = 1, \dots, n_S - 2. \quad (5-9)$$

Bild 5-3 veranschaulicht diese Vorgehensweise für ein sehr einfaches Beispiel mit insgesamt vier Stützstellen, von denen die beiden inneren neu berechnet werden.

Die Neuverteilung bestehender Knoten für die skalare Spline-Interpolation lässt sich leicht auf die für diese Arbeit erforderliche matrixwertige Interpolation gemäß Gleichung (5-6) erweitern. Die Approximation der Krümmung erfolgt dazu

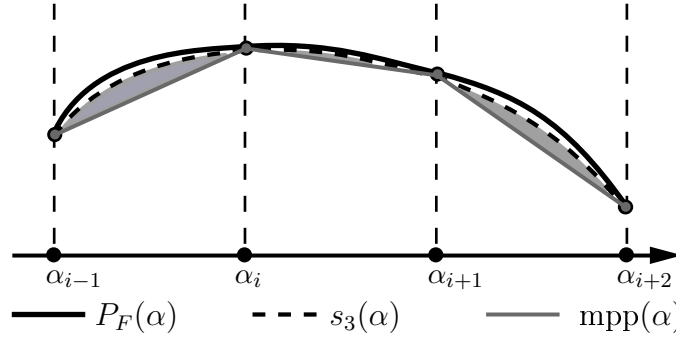


Bild 5-4: Schematische Darstellung der Berechnung von Δ zur Bestimmung der Approximationsgüte von $mpp(\alpha)$

komponentenweise durch die Funktionen $h_{i,j}$, $1 \leq i, j \leq n_x$. In der Funktion H wird dann die maximale Krümmung aller Komponenten verwendet, d. h.

$$H(z) = \int_0^z (\max_{i,j} h_{i,j}(\tau))^{1/2} d\tau, \quad 1 \leq i, j \leq n_x.$$

Hierdurch wird gewährleistet, dass die Stützstellen dichter platziert werden, sobald in mindestens einer Komponente des linearen Splines mpp eine hohe Krümmung vorliegt.

Die Konvergenz des *newnot*-Algorithmus kann nicht garantiert werden. Aufgrund der Annahmen und Vereinfachungen existieren sogar Beispiele, bei denen die Neuverteilung der Stützstellen zu einer Verschlechterung der Approximation führt, siehe [Boo01] für weitere Ausführungen zu dieser Problematik. Für eine automatisierte Festlegung der Stützstellen zur Interpolation der paretooptimalen Systeme ist daher die hier vorgestellte Variante des *newnot*-Algorithmus allein nicht ausreichend. Es wird ergänzend ein weiteres Kriterium zur Bestimmung der Approximationsgüte der linearen Interpolation verwendet.

Um den Rechenaufwand möglichst gering zu halten, wird die lineare Spline-Interpolation der paretooptimalen Systeme mit einer kubischen, ebenfalls matrixwertigen Spline-Interpolation

$$s_3 : [0, \alpha_{max}] \rightarrow \mathbb{R}^{n_x \times n_x}, \quad \alpha \mapsto s_3(\alpha)$$

verglichen. Die kubische Spline-Interpolation beruht dabei auf denselben Stützstellen, sodass keine zusätzlichen Auswertungen der Systemmatrizen durchgeführt werden müssen. Als Kriterium wird die relative quadratische Fehlerfläche zwi-

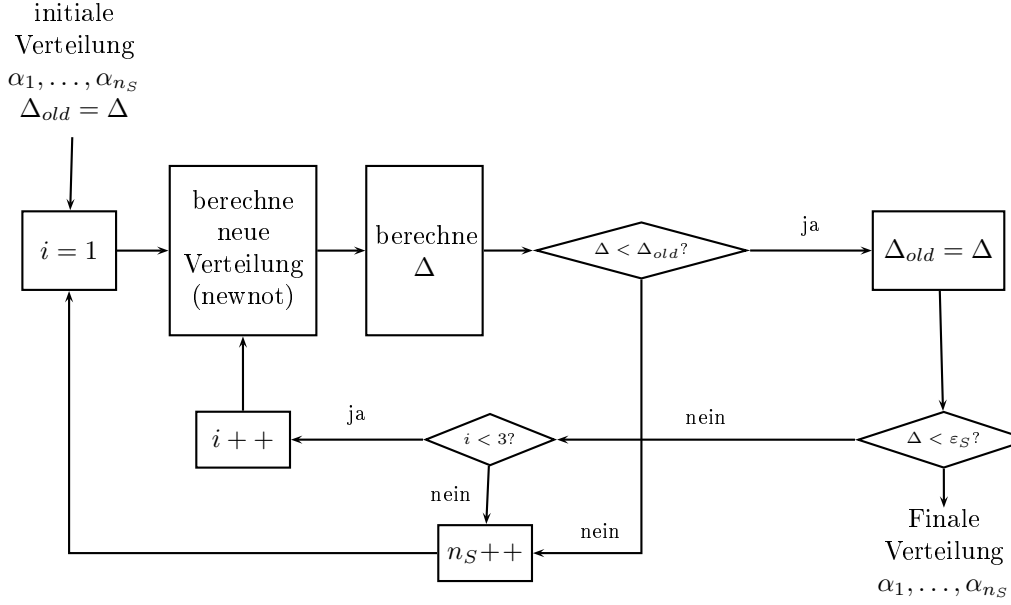


Bild 5-5: Aktivitätsdiagramm des Algorithmus zur automatisierten Diskretisierung der paretooptimalen Systeme

schen den beiden Splines betrachtet, die in Bild 5-4 schematisch dargestellt ist. Dazu wird zunächst komponentenweise die \mathcal{L}^2 -Norm

$$\begin{aligned}
 L_{i,j} &:= \left\| (\text{mpp}(\alpha) - s_3(\alpha))_{i,j} \right\|_{\mathcal{L}^2} \\
 &= \left(\int_0^{\alpha_{\max}} (\text{mpp}(\alpha) - s_3(\alpha))_{i,j}^2 d\alpha \right)^{1/2}, \quad 1 \leq i, j \leq n_x,
 \end{aligned}$$

berechnet. Die resultierenden Ergebnisse $L_{i,j}$ können in einer $n_x \times n_x$ -Matrix L zusammengefasst werden. Um einen skalaren Wert der Abweichung zwischen mpp und s_3 zu erhalten, wird die 1-Norm von L bestimmt. Beide Schritte, die komponentenweise \mathcal{L}^2 -Norm sowie die Matrix 1-Norm, werden im Folgenden mit $\|\cdot\|_{1 \circ \mathcal{L}^2}$ bezeichnet. Als endgültiges Kriterium wird die relative Abweichung

$$\Delta := \frac{\|(\text{mpp}(\alpha) - s_3(\alpha))\|_{1 \circ \mathcal{L}^2}}{\|(\text{mpp}(\alpha))\|_{1 \circ \mathcal{L}^2}} \quad (5-10)$$

verwendet. Da es sich bei den Splines um stückweise definierte Polynome handelt, kann die Fehlerfläche algorithmisch mit geringem Rechenaufwand bestimmt werden.

Der vollständige Algorithmus für die automatisierte Knotenverteilung ist in Bild 5-5 dargestellt. Als Eingangsdaten sind eine initiale Verteilung $\alpha_1, \dots, \alpha_{n_S}$, die

beispielsweise einer äquidistanten Zerlegung entsprechen kann, und die zugehörige Fehlerfläche Δ_{old} , berechnet gemäß (5-10), erforderlich. Weiterhin muss eine Abbruchschranke ε_S vorgegeben werden. Zu Beginn werden dann mit Hilfe des beschriebenen *newnot*-Algorithmus eine neue Verteilung der Stützstellen und die resultierende Fehlerfläche Δ bestimmt. Ist die neue Fehlerfläche kleiner als bisher, liegt also eine verbesserte Approximation vor, so wird die neue Verteilung übernommen und geprüft, ob die vorgegebene Fehlerschranke ε_S unterschritten wurde. Ist dies nicht der Fall, wird erneut eine neue Verteilung mit gleicher Stützstellenzahl berechnet. Dies geschieht bis zu drei Mal hintereinander.

Ist die Abbruchschranke dann nicht erreicht oder wurde eine Knotenverteilung bestimmt, die zu einer schlechteren Approximationsgüte Δ führt, so wird die Anzahl der Stützstellen n_S erhöht. Dies ist sehr einfach in Kombination mit dem *newnot*-Algorithmus zu implementieren, da lediglich Gleichung (5-9) mit der erhöhten Stützstellenzahl $n_S + 1$ auszuwerten ist.

Die Konvergenz des hier vorgestellten Algorithmus ist gesichert unter der Annahme, dass die zu interpolierenden paretooptimalen Systeme $A(s(\alpha))$ eine hinreichend glatte Funktion darstellen. Alle Komponenten sowohl der linearen Interpolation mpp als auch der kubischen Interpolation s_3 konvergieren gleichmäßig gegen die zu approximierende Funktion, falls die folgenden Bedingungen erfüllt sind. Zum einen muss die zu approximierende Funktion entsprechend oft stetig differenzierbar sein. Zum anderen muss die Feinheit¹ der Stützstellenverteilung mit zunehmender Stützstellenzahl abnehmen, siehe z. B. [Boo01] oder [Wer92] für entsprechende Fehlerabschätzungen. Sind diese Bedingungen für alle Komponenten der matrixwertigen Interpolation erfüllt, konvergiert auch das Approximationsmaß Δ aus (5-10) gegen Null.

Allerdings ist durch die Verwendung des *newnot*-Algorithmus zur Platzierung der Stützstellen nicht gewährleistet, dass die Feinheit der Zerlegung mit steigender Stützstellenzahl abnimmt. In Bereichen, in denen die zu interpolierende Funktion einen linearen Verlauf aufweist, nimmt die Approximation h der Krümmung, die in (5-8) verwendet wird, den Wert null an. Dies führt dazu, dass H nicht mehr streng monoton wachsend, sondern stückweise konstant ist. Die Umkehrfunktion H^{-1} ist streng genommen an diesen Punkten nicht definiert. Bei der numerischen Umsetzung werden derartige Fälle gesondert abgefangen. Eine Verfeinerung der Zerlegung findet in diesen Bereichen jedoch ggf. nicht statt. Bild 5-6 zeigt diesen Sonderfall für eine speziell konstruierte Testfunktion, die zwischen π und 2π konstant ist. Ausgehend von 5 äquidistanten Stützstellen, zu finden im linken Teil von Bild 5-6, wird die Anzahl sukzessive unter Verwendung des *newnot*-Algorithmus erhöht. Im rechten Teil von Bild 5-6 ist die Verteilung bei 23 Stütz-

¹Die Feinheit δ einer Zerlegung des Intervalls $[a, b]$ mit $a = \alpha_1 < \alpha_2 < \dots < \alpha_{n_S} = b$ ist gegeben als $\delta = \max_{1 \leq i < n_S} (\alpha_{i+1} - \alpha_i)$.

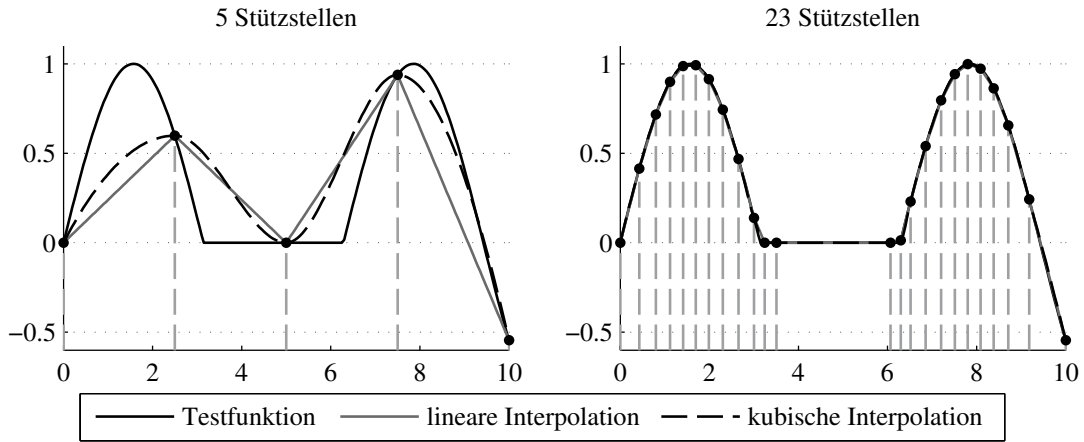


Bild 5-6: Einfaches Beispiel zur Illustration der Stützstellenverteilung nach wiederholter Ausführung des newnot-Algorithmus

stellen dargestellt. Es ist deutlich zu erkennen, dass die Feinheit der Zerlegung nicht abgenommen hat. Allerdings stimmen gerade in Bereichen, in denen die zu interpolierende Funktion einen konstanten oder linearen Verlauf aufweist, die lineare und kubische Interpolation ohnehin überein, sodass die Konvergenz des Kriteriums Δ trotzdem gewährleistet ist.

Zusammenfassend lässt sich festhalten, dass nach der Anwendung der vorgestellten Interpolation Systeme mit abschnittsweise definierten Dynamikmatrizen vorliegen, die über eine affin-lineare Parameterabhängigkeit verfügen. Eine Anwendung der parametrischen Modellordnungsreduktion ist somit möglich. Die Interpolation erfolgt weitgehend automatisiert durch den vorgestellten Algorithmus. Lediglich eine initiale Stützstellenverteilung muss erstellt werden. Zudem muss die Fehlerschranke ε_S systemabhängig angepasst werden.

5.3 Parametrische Modellordnungsreduktion paretooptimaler Systeme

Durch die Parametrierung und die anschließende Interpolation der paretooptimalen Systeme sind die Voraussetzungen erfüllt, um die parametrische Modellordnungsreduktion anwenden zu können. Nach der Durchführung der in den beiden vorhergehenden Abschnitten beschriebenen Schritte sind die zur Paretomenge gehörenden dynamischen Systeme gegeben als

$$\begin{aligned} \dot{x}(t) &= \text{mpp}(\alpha)x(t) + Bu(t), \\ y(t) &= Cx(t), \end{aligned} \tag{5-11}$$

mit der matrixwertigen und stückweise linearen Funktion mpp aus (5-6).

Die Modellordnungsreduktion kann nun sowohl mit der manuellen rationalen Interpolation, als auch mit der Matrix Interpolation erfolgen. Bei der manuellen rationalen Interpolation muss jeder Abschnitt von (5-11) separat betrachtet werden. Das zugehörige parametrische System

$$\begin{aligned}\dot{x}(t) &= (A_i + \bar{\alpha}_i \Delta A_i) x(t) + Bu(t), \\ y(t) &= Cx(t), \quad i = 1, \dots, n_S - 1,\end{aligned}$$

verfügt über die erforderliche affin-lineare Parameterabhängigkeit der Dynamikmatrix. Mit Hilfe des in Abschnitt 3.1.2 beschriebenen objektorientierten Arnoldi-Algorithmus können daher die beiden Projektionsmatrizen V_i und W_i bestimmt werden. Das reduzierte System eines Abschnitts ist dann gemäß (3-5) gegeben als

$$\begin{aligned}W_i^T V_i \dot{x}_r(t) &= (W_i^T A_i V_i + \bar{\alpha}_i W_i^T \Delta A_i V_i) x_r(t) + W_i^T Bu(t), \\ y_r(t) &= C V_i x_r(t).\end{aligned}$$

Jedes reduzierte System für sich betrachtet ist stetig bzgl. α , die Stetigkeit geht jedoch an den Intervallgrenzen verloren. Für eine beliebige Stützstelle α_{i+1} gilt $\bar{\alpha}_i = 1$ und $\bar{\alpha}_{i+1} = 0$. Die zugehörigen reduzierten Dynamikmatrizen der beiden Abschnitte sind demnach an der Stützstelle α_{i+1} gegeben als

$$W_i^T (A_i + 1 \cdot \Delta A_i) V_i = W_i^T A_{i+1} V_i$$

für Abschnitt i und

$$W_{i+1}^T (A_{i+1} + 0 \cdot \Delta A_{i+1}) V_{i+1} = W_{i+1}^T A_{i+1} V_{i+1}$$

für den angrenzenden Abschnitt $i + 1$. Diese beiden Matrizen werden im allgemeinen voneinander verschieden sein. Da es sich bei der manuellen rationalen Interpolation generell um ein lokales Verfahren handelt, bei dem nur ein spezielles parametrisches System berücksichtigt werden kann, besteht keine Möglichkeit, die aufgezeigte Problematik zu beheben. Eine Anwendung der manuellen rationalen Interpolation ist dennoch möglich, da zumindest bei den untersuchten Anwendungsfällen die Abweichungen zwischen den unterschiedlichen reduzierten Systemen keinen signifikanten Einfluss auf die weitere Verwendung des parametrischen reduzierten Systems hatten.

Zur Anwendung der Matrix Interpolation werden die zu den Stützstellen α_i gehörenden Systeme

$$\begin{aligned}\dot{x}(t) &= A_i x(t) + Bu(t), \\ y(t) &= Cx(t),\end{aligned}$$

separat reduziert. Die dafür erforderlichen Projektionsmatrizen V_i und W_i können prinzipiell mit einem beliebigen Reduktionsverfahren bestimmt werden. Im Rahmen dieser Arbeit wird dazu stets die \mathcal{H}_2 -optimale Interpolation, eingeführt in Abschnitt 2.4, benutzt.

Die Interpolation der resultierenden reduzierten Systeme

$$\begin{aligned} W_i^T \dot{x}_r(t) &= W_i^T A_i V_i x_r(t) + W_i^T B u(t), \\ y_r(t) &= C V_i x(t), \end{aligned}$$

erfolgt gemäß der Interpolationsvorschrift aus (3-15) unter Verwendung der Gewichtsfunktionen ω_i wie sie im vorhergehenden Abschnitt in (5-7) definiert wurden. Die hierfür notwendigen Matrizen M_i und T_i für die Rückprojektion auf einen gemeinsamen Unterraum, siehe (3-11) und (3-12) in Abschnitt 3.2, können dabei auf zwei Wegen bestimmt werden. Einerseits kann, analog zur manuellen rationalen Interpolation, jeder Abschnitt für sich betrachtet werden. Die zur Berechnung von M_i und T_i benötigte Matrix R wird dann aus der Singulärwertzerlegung der Matrix

$$V_{all,i} = [V_i, V_{i+1}], \quad i = 1, \dots, n_S - 1, \quad (5-12)$$

berechnet. Diese abschnittsweise Interpolation führt jedoch ebenfalls zu einem Verlust der Stetigkeit an den Intervallgrenzen.

Andererseits kann die Matrix R auch basierend auf allen Projektionsmatrizen, d. h. mit Hilfe von

$$V_{all} = [V_1, \dots, V_{n_S}],$$

bestimmt werden. In diesem Fall findet eine Rückprojektion aller reduzierten Systeme auf einen einzigen gemeinsamen Unterraum statt. Ein stetiger Übergang zwischen den Intervallen ist dann gewährleistet, weshalb diese Variante der Matrix Interpolation zu bevorzugen ist. Eine Voraussetzung hierfür ist allerdings, dass alle reduzierten Systeme die gleiche Systemordnung q aufweisen. Zudem können Probleme auftreten, wenn sich die einzelnen Unterräume, aufgespannt von den Matrizen V_i , stark unterscheiden. Dann führt eine Rückprojektion auf den gemeinsamen Unterraum eventuell zu einer deutlichen Verschlechterung der Approximationsgüte. Wie stark die einzelnen Unterräume voneinander abweichen lässt sich aus der Singulärwertzerlegung von V_{all} schließen. Ähneln die Unterräume einander, so ist ein deutlicher Abfall zwischen den Singulärwerten an den Stellen q und $q + 1$ erkennbar.

Die Stabilität der reduzierten Systeme stellt neben der Stetigkeit eine weitere wichtige Eigenschaft dar, die erfüllt sein muss. Hier besteht das Hauptproblem

darin, dass für Verfahren der Modellordnungsreduktion, die auf einer Interpolation der Übertragungsfunktion basieren, in den meisten Fällen kein Erhalt der Stabilität garantiert werden kann. Sowohl bei der manuellen rationalen Interpolation, als auch bei der \mathcal{H}_2 -optimalen Interpolation können prinzipiell instabile reduzierte Systeme berechnet werden. Es existieren Ansätze, mit denen die Stabilität garantiert werden kann, indem beispielsweise zusätzliche lineare Matrixungleichungen (LMI) gelöst werden, siehe [BD08]. Dieser Ansatz wurde im Rahmen der vorliegenden Arbeit untersucht, erwies sich aber für die betrachteten Anwendungsfälle als unbrauchbar. Die LMI ließen sich entweder mit den angegebenen Algorithmen nicht lösen oder die berechneten Lösungen führten nicht zu einer Stabilisierung des reduzierten Systems. Andere Ansätze wie z. B. der bereits erwähnte Artikel [ECSP⁺11], verwenden einseitige Projektionen an Stelle zweiseitiger, um die Stabilität zu garantieren. Dies führt zu einer Verschlechterung der Approximationsgüte bei gleicher Ordnung q des reduzierten Modells. Innerhalb dieser Arbeit wird daher die Stabilität manuell überprüft und gegebenenfalls Anpassungen an den Algorithmus-Parametern der Reduktionsverfahren vorgenommen, um stabile reduzierte Systeme zu erhalten.

Bei der Matrix Interpolation entsteht durch die Interpolation der Systemmatrizen eine zusätzliche Problematik hinsichtlich der Stabilität. Selbst bei stabilen reduzierten Systemen an den Stützstellen, ist nicht garantiert, dass die gewichtete Summe der Systemmatrizen in jedem Fall zu stabilen Systemen führt. Allerdings trat diese Problematik bei den betrachteten Anwendungsbeispielen nicht auf. Unter Umständen ist dies auf den speziellen Einsatzfall der Interpolation paretooptimaler Systeme zurückzuführen. Die paretooptimalen Systeme selbst sind stabil, da sie andernfalls nicht paretooptimal wären. Die in den durchgeführten Optimierungen beobachtete Streuung der Paretopunkte im Urbildraum veranschaulicht, dass die Systeme auch bei kleineren Abweichungen der Parameter stabil bleiben. Da die lineare Interpolation der paretooptimalen Systeme eine komponentenweise Näherung dieser Parameterabhängigkeit darstellt, sind dies Indizien dafür, dass stets stabile Systeme zu erwarten sind. In Ergänzung zu dem hier vorgestellten Ansatz könnte zukünftig mit Hilfe gemeinsamer Lyapunov-Funktionen, siehe [Wul04], die Stabilität der interpolierten Systeme nachgewiesen werden. Mit diesem Ansatz wurde beispielsweise in [KDL08, KPL09] die Stabilität einer adaptiven LQ-Regelung gezeigt.

Die reduzierten paretooptimalen Systeme, die das Resultat der parametrischen Modellordnungsreduktion bilden, sind in Tabelle 5-1 einander gegenübergestellt. Bei der manuellen rationalen Interpolation bleibt die Struktur des Systems (5-11), bestehend aus einer parameterabhängigen Dynamikmatrix und konstanten Ein- und Ausgangsmatrizen, erhalten. Die Matrix Interpolation führt demgegenüber zu einer Parameterabhängigkeit aller Matrizen.

Tabelle 5-1: Endergebnis der parametrischen Modellordnungsreduktion paretooptimaler Systeme

Reduzierte paretooptimale Systeme	
manuelle Interpolation	Matrix Interpolation
$E_{r,i}\dot{x}_r = A_{r,i}(\alpha)x_r + B_{r,i}u$ $y_r = C_{r,i}x_r, \alpha \in [\alpha_i, \alpha_{i+1})$	$E_r(\alpha)\dot{x}_r = A_r(\alpha)x_r + B_r(\alpha)u$ $y_r = C_r(\alpha)x_r$
$E_{r,i} = W_i^T V_i,$	$E_r(\alpha) = \sum_{i=1}^{n_S} \omega_i(\alpha) M_i E_{r,i} T_i^{-1}$
$A_{r,i}(\alpha) = W_i^T (A_i + \bar{\alpha}_i \Delta A_i) V_i$	$A_r(\alpha) = \sum_{i=1}^{n_S} \omega_i(\alpha) M_i A_{r,i} T_i^{-1}$
$B_{r,i} = W_i^T B$	$B_r(\alpha) = \sum_{i=1}^{n_S} \omega_i(\alpha) M_i B_{r,i}$
$C_{r,i} = C V_i$	$C_r(\alpha) = \sum_{i=1}^{n_S} \omega_i(\alpha) C_{r,i} T_i^{-1}$

Für die weitere Verwendung der reduzierten paretooptimalen Systeme ist bei beiden Verfahren von Vorteil, dass eine aufwändige Erzeugung des Systems bei Parameteränderungen, z.B. durch die Linearisierung eines Simulinkmodells, entfällt. Es ist lediglich erforderlich, die Systemmatrizen der reduzierten Systeme sowie die Knotenpunkte α_i rechnerintern abzulegen und mit Hilfe geeigneter Funktionen auszuwerten. Insbesondere im Kontext der hierarchischen Strukturierung bleibt so die Kapselung unterlagerter Systeme erhalten und es ist kein spezielles Systemwissen erforderlich, um eine Parameteränderung vorzunehmen. Durch die an den Zielfunktionen orientierte Parametrierung ist eine abstrakte und domänenunabhängige Variation der Systemparameter möglich. Die Parametrierung ist darüber hinaus für die hierarchische Optimierung und selbstoptimierende Regelungen geeignet, was die Anwendungen in Kapitel 6 verdeutlichen.

5.4 Resultate am Feder-Neige-Prüfstand

Basierend auf den Ergebnissen der Mehrzieloptimierung aus Abschnitt 4.4 wird in diesem Abschnitt die PMOR paretooptimaler Systeme an zwei Anwendungsbeispielen demonstriert. Beide Anwendungen gehören zum Feder-Neige-Prüfstand. Zum einen wird in Abschnitt 5.4.1 das Aktormodul als Teilsystem des Prüfstands reduziert. Zum anderen wird in Abschnitt 5.4.2 der gesamte Prüfstand betrach-

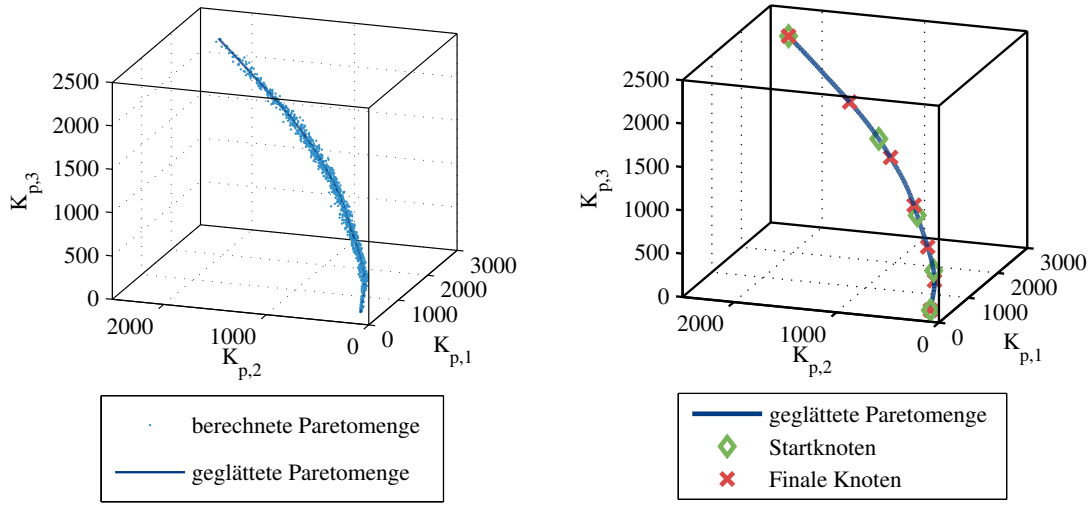


Bild 5-7: links: Mit GAIO berechnete Paretomenge des Aktormoduls und zugehöriger Ausgleichsspline; rechts: äquidistante Stützstellen als Startwerte der Knotenverteilung und resultierende Stützstellen für die PMOR

tet. Für die Reduktion kommen sowohl die manuelle rationale Interpolation als auch die Matrix Interpolation zum Einsatz.

5.4.1 PMOR des Aktormoduls des Feder-Neige-Prüfstands

Die PMOR des Aktormoduls basiert auf den Ergebnissen der Mehrzieloptimierung, die im vorangegangenen Kapitel beschrieben sind. Das reduzierte Modell wird im nächsten Kapitel zur hierarchischen Optimierung des Feder-Neige-Prüfstands verwendet. Die Grundlage hierfür bildet das hierarchische Modell, auf dessen Struktur in Abschnitt 1.3 eingegangen wird. Zur Reduktion wird daher nicht das Optimierungsmodell des Aktormoduls, sondern ein lineares Modell verwendet, das das dynamische Verhalten des geregelten Systems abbildet.

Die in Bild 4-6 abgebildete Paretomenge stellt das Ergebnis der Mehrzieloptimierung des Aktormoduls dar und bildet die Grundlage für die PMOR. Da die Paretopunkte eine erhebliche Streuung aufweisen wird als erster Schritt ein kubischer *smoothing spline* zur Glättung der Paretomenge gemäß (5-4) berechnet. Dieser ist in Bild 5-7 links zusammen mit den berechneten Paretopunkten dargestellt. Als Parametrisierung für die PMOR des Aktormoduls wird der von den Minima der beiden Zielfunktionen gebildete Simplex gewählt, siehe (5-3). Als Maximalwert wird $\alpha_{max} = 1000$ gesetzt.

Für die Interpolation der paretooptimalen Systeme wird der in Bild 5-5 dargestellte Algorithmus genutzt. Initial werden dazu fünf äquidistante Stützstellen $\alpha_i, i = 1, \dots, 5$, sowie eine Abbruchschranke $\varepsilon_S = 10^{-10}$ gewählt. Die initiale

Stützstellenverteilung ist in Bild 5-7 rechts zu finden. Es ist deutlich zu erkennen, dass die äquidistante Verteilung bzgl. α nicht zu einer äquidistanten Verteilung im Urbild- bzw. Parameterraum führt. Als Ergebnis der automatisierten Interpolation der paretooptimalen Systeme erhält man sieben Stützstellen, die ebenfalls in Bild 5-7 rechts eingetragen sind. Die neue Verteilung besitzt dabei deutlich gleichmäßigere Abstände im Urbildraum.

Nachfolgend werden die durch α parametrisierten paretooptimalen Systeme des Aktormoduls durch Anwendung der manuellen rationalen Interpolation (siehe Abschnitt 3.1) reduziert. Die Reduktion erfolgt auf eine Systemordnung von $q = 9$, ausgehend von der ursprünglichen Systemordnung 18. Jeder der sechs Abschnitte $[\alpha_i, \alpha_{i+1}]$ wird separat betrachtet. Für den objektorientierten Arnoldi-Algorithmus werden jeweils die drei Entwicklungspunkte $s_0 \in \{0, 90, 250\}$ mit je drei korrespondierenden Basisvektoren verwendet. Die Schranke für die Deflation beträgt $\varepsilon_D = 10^{-4}$. Die Deflation kommt für den konkreten Anwendungsfall mit dieser Wahl nicht zum Tragen; die drei berechneten Projektionsvektoren sind jeweils linear unabhängig. Auch die gesamten Projektionsmatrizen $V = [V_1, V_2, V_3]$ und $W = [W_1, W_2, W_3]$, die aus den zu den drei Entwicklungspunkten gehörenden Projektionsmatrizen gebildet werden, haben in diesem Fall vollen Rang.

Der resultierende Reduktionsfehler für den gesamten Parameterbereich $[0, \alpha_{max}]$ ist in Bild 5-8 aufgetragen. Die bei der Interpolation der paretooptimalen Systeme bestimmten Stützstellen sind besonders hervorgehoben. Zur Bewertung der Approximationsgüte werden die relativen Fehler bzgl. der \mathcal{H}_2 sowie der \mathcal{H}_∞ -Norm herangezogen (siehe Abschnitt 2.2). Dabei werden die parametrischen reduzierten Systeme mit den ursprünglichen linearen Systemen verglichen, d. h. zu einem gegebenen Wert von α werden die zugehörigen Reglerparameter berechnet und daraus das jeweilige lineare System bestimmt.

Die Abweichungen liegen in fast allen Bereichen deutlich unterhalb von 10%, was ein sehr gutes Reduktionsergebnis darstellt. Der Approximationsfehler steigt mit wachsendem α an. Steigende Werte von α entsprechen dabei Systemkonfigurationen, die zu weniger Energieverbrauch und einer langsameren Systemdynamik führen. Dies wirkt sich anscheinend negativ auf den Reduktionsfehler aus, wobei die genaue Ursache nicht angegeben werden kann. Durch die abschnittsweise Berechnung der reduzierten Systeme entstehen die Sprünge im Reduktionsfehler an den Stützstellen. Dies veranschaulicht die fehlende Stetigkeit der reduzierten Systeme bzgl. α .

5.4.2 PMOR des gesamten Feder-Neige-Prüfstands

Als zweites Anwendungsbeispiel der Reduktion paretooptimaler Systeme wird der gesamte Feder-Neige-Prüfstand betrachtet. Analog zum Aktormodul dient auch für den Feder-Neige-Prüfstand die im vorhergehenden Kapitel beschriebene

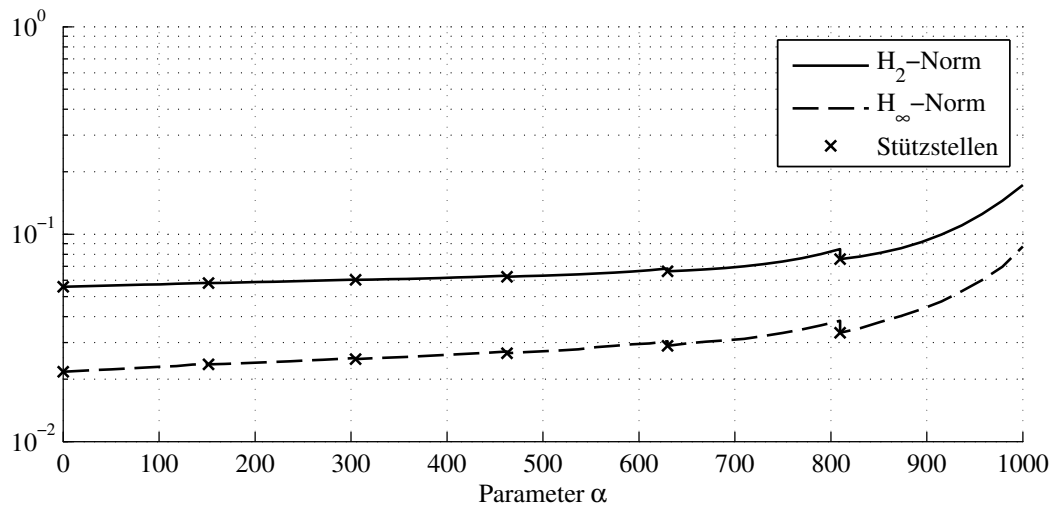


Bild 5-8: Relative Fehler der PMOR des Aktormoduls für den gesamten Parameterbereich

Mehrzieloptimierung als Grundlage für die PMOR. Im Unterschied zum Aktormodul wird beim Feder-Neige-Prüfstand kein gesondertes Modell, sondern direkt der lineare Teil des Optimierungsmodells, vgl. Bild 5-1, reduziert.

Als erster Schritt wird die Paretomenge parametrisiert. Hierzu werden beide vorgestellten Varianten verwendet, der Simplex zwischen den Minima der beiden Zielfunktionen (5-3) sowie der Winkel des Zielfunktionsvektors (5-2). Da auch beim Feder-Neige-Prüfstand die vom Optimierungsalgorithmus berechneten Paretopunkte eine relativ hohe Streuung aufweisen, wird im Weiteren ein *smoothing spline* an Stelle der ursprünglichen Paretopunkte genutzt.

Für die Interpolation der paretooptimalen Systeme mit dem Algorithmus aus Bild 5-5 werden initial für beide Parametrisierungen 10 äquidistante Stützstellen gewählt und die Abbruchschranke $\varepsilon_S = 10^{-6}$ verwendet. Die resultierenden Stützstellenverteilungen sind in Bild 5-9 zu finden. Für die Parametrisierung mit dem Simplex werden vom Algorithmus 26 Stützstellen berechnet, bei der Verwendung des Winkels sind es 29.

Die Matrix Interpolation (siehe Abschnitt 3.2) wird zur PMOR der paretooptimalen Systeme genutzt. Dazu werden die Systeme an den Stützstellen mit Hilfe der \mathcal{H}_2 -optimalen Interpolation aus Abschnitt 2.4 reduziert. Die Reduktion erfolgt automatisiert, indem der IRKA mit zufälligen initialen Interpolationspunkten an jeder Stützstelle solange wiederholt wird bis die resultierende Abweichung zwischen reduziertem und nicht reduziertem System bzgl. der \mathcal{H}_2 -Norm kleiner als 0.03 ist. Die nicht reduzierten, paretooptimalen Systeme besitzen die Systemordnung 98 und werden alle auf Ordnung $q = 20$ reduziert.

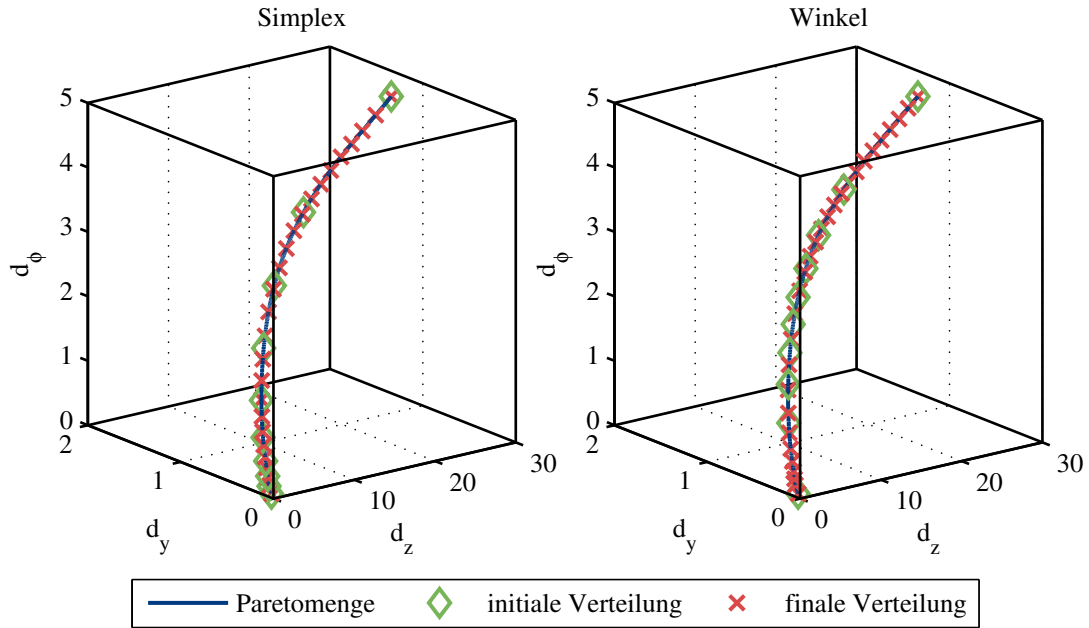


Bild 5-9: Stützstellenverteilung der Interpolation paretooptimaler Systeme für den Feder-Neige-Prüfstand

Der Reduktionsfehler für den gesamten Parameterbereich ist in Bild 5-10 dargestellt. Generell liegt ein sehr gutes Reduktionsergebnis vor. Der Fehler liegt sowohl bzgl. der \mathcal{H}_2 -Norm, als auch bzgl. der \mathcal{H}_∞ -Norm in fast allen Bereichen deutlich unter 10%. Dabei ist die \mathcal{H}_2 -Norm stets kleiner als die \mathcal{H}_∞ -Norm. Für beide Parametrisierungen ist zu beobachten, dass der Reduktionsfehler an den Stützstellen mit wachsendem α ansteigt. Ansteigende α -Werte entsprechen hierbei einer ansteigenden Dämpfung durch die aktive Federung. An den Stützstellen α_i gleicht der Reduktionsfehler der Matrix-Interpolation dem Fehler der nichtparametrischen Reduktion. Dies folgt direkt aus der Wahl der Gewichtsfunktionen ω_i in (5-7).

Der Reduktionsfehler zwischen zwei Stützstellen unterscheidet sich im vorderen Bereich qualitativ vom Reduktionsfehler im hinteren Bereich. Dies gilt für beide Parametrisierungen gleichermaßen. Für kleine α -Werte steigt der Reduktionsfehler zur Intervallmitte hin stark an, während er bei größeren α -Werten über das gesamte Intervall hinweg in etwa gleich bleibt. Diese beiden Varianten gehen schlagartig ineinander über. Beim Simplex ist der Übergang etwa bei $\alpha = 550$ zu erkennen. Bei der Parametrisierung mit dem Winkel liegt der Übergang bei etwa $\alpha = 140$. Die zugehörigen Systeme sind jedoch fast identisch. Dies zeigt Bild 5-11, in dem die Stützstellen der kritischen Intervalle auf der Pareto Menge eingezeichnet sind. Als kritisches Intervall ist hier der Bereich $[\alpha_i, \alpha_{i+1}]$ gemeint, in dem

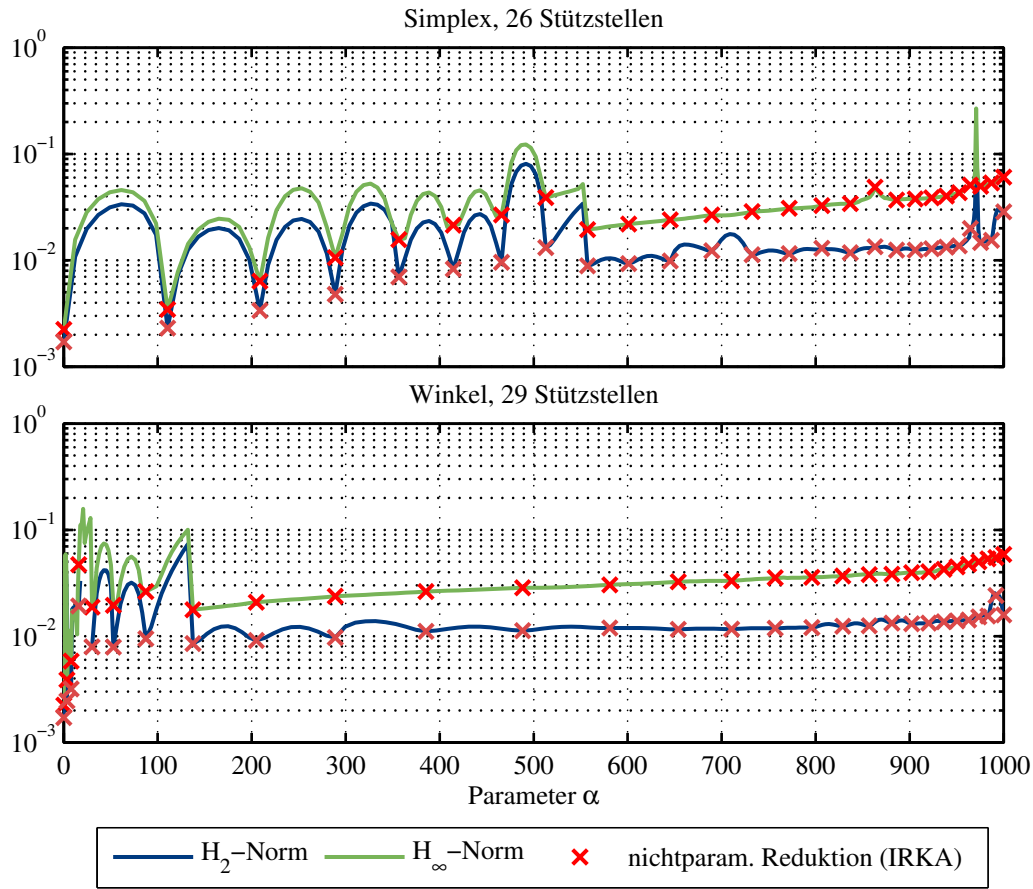


Bild 5-10: Relative Fehler der PMOR des Feder-Neige-Prüfstands für den gesamten Parameterbereich und beide Parametrierungsvarianten

der Übergang liegt. Die unterschiedlichen Werte für α sind auf die geometrischen Eigenschaften der Parametrierung zurückzuführen und nicht durch die PMOR bedingt.

Insbesondere bei der Parametrierung mit dem Winkel wird deutlich, dass der stark unterschiedliche Reduktionsfehler nicht mit der Intervalllänge zwischen zwei Stützstellen zusammenhängt. Die schlechtere Approximation im Anfangsbereich ist vielmehr auf die Rückprojektion auf den gemeinsamen Unterraum mittels der Matrix R zurückzuführen, siehe (3-11) und (3-12). Die dort benötigte Matrix R wird abschnittsweise aus einer Singulärwertzerlegung von $V_{all,i} = [V_i, V_{i+1}]$ bestimmt, vgl. auch (5-12).

Einige der Singulärwerte dieser Zerlegung sind exemplarisch für drei Intervalle der Simplex-Parametrierung in Bild 5-12 dargestellt. Das erste Intervall $[\alpha_7, \alpha_8]$ repräsentiert hierbei ein Intervall mit stark anwachsendem Reduktionsfehler. Das

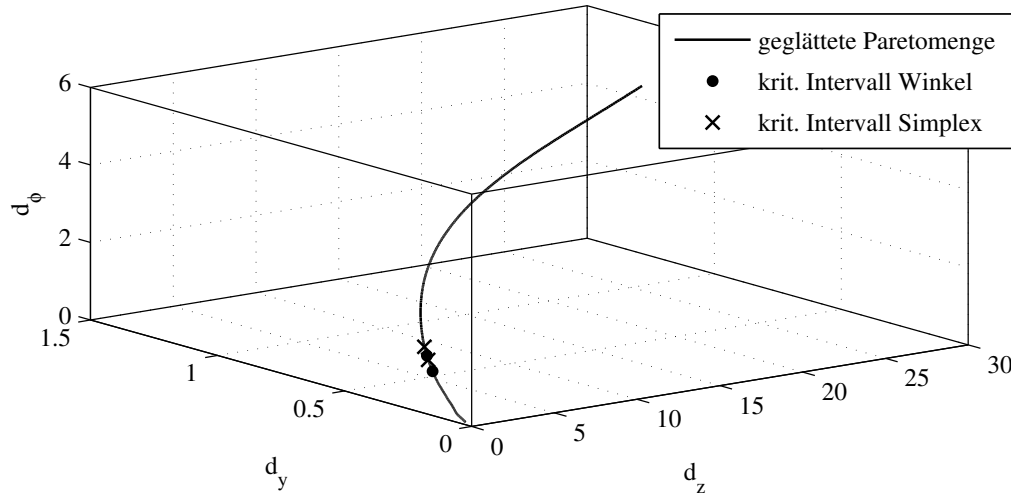


Bild 5-11: Lage der kritischen Intervalle im Urbildraum für beide Parametrierungen

zweite Intervall $[\alpha_8, \alpha_9]$ ist das kritische Intervall und das dritte hier dargestellte Intervall $[\alpha_9, \alpha_{10}]$ steht stellvertretend für die Intervalle mit kaum ansteigendem Reduktionsfehler.

Bei dem letztgenannten ist ein sehr deutlicher Abfall der Singulärwerte zwischen dem 20. und 21. Singulärwert zu erkennen. Dies zeigt, dass die von den Matrizen V_9 und V_{10} aufgespannten Unterräume sehr ähnlich sind. Daraus wiederum folgt, dass die Rückprojektion auf den gemeinsamen Unterraum sehr gute Ergebnisse liefert. Anders sieht dies bei den beiden übrigen Intervallen aus. Dort ist der Abfall der Singulärwerte deutlich langsamer. Insbesondere bei dem kritischen Intervall sind der 20. und 21. Singulärwert annähernd gleich groß. Für die Rückprojektion mit der Matrix R wird jedoch nur einer der beiden Singulärwerte berücksichtigt. Hierauf ist der größere Reduktionsfehler bei der Matrix Interpolation zurückzuführen, der trotz einer sehr guten Approximation der nicht-parametrischen Reduktion an den Stützstellen auftritt. Da auf die Spalten der Matrix V bei Verwendung des IRKA keinerlei Einfluss genommen werden kann, lässt sich diese Problematik mit den in dieser Arbeit behandelten Methoden nicht beheben.

Die Berechnung einer gemeinsamen Matrix R für alle Intervalle ist ebenfalls untersucht worden, führt aber zu einem parametrischem reduzierten Modell mit einer sehr schlechten Approximationsgüte. Dies bedeutet für das konkrete Anwendungsbeispiel, dass auch bei der Matrix Interpolation keine Stetigkeit bzgl. α über die Intervallgrenzen hinaus hergestellt werden kann.

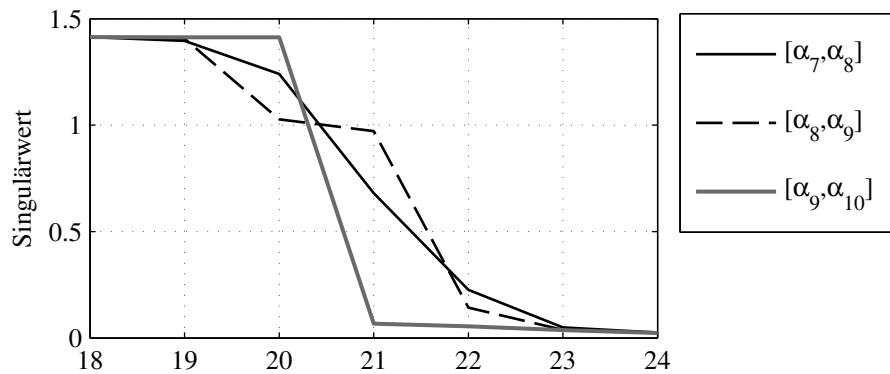


Bild 5-12: Singulärwerte aus der Singulärwertzerlegung von $[V_i, V_{i+1}]$ für drei Intervalle

Abschließend sei noch angemerkt, dass auch die manuelle rationale Interpolation für die Reduktion des gesamten Feder-Neige-Prüfstands untersucht wurde. Es konnten allerdings keine geeigneten Parameter (Ordnung des reduzierten Systems, Entwicklungspunkte, etc.) gefunden werden, die zu einem zufriedenstellenden Reduktionsergebnis führen.

5.5 Erweiterungsmöglichkeiten

Die in diesem Kapitel beschriebene Methode zur PMOR paretooptimaler Systeme konnte erfolgreich an den im vorhergehenden Abschnitt 5.4 dargestellten Anwendungsbeispielen angewendet werden. Die resultierenden reduzierten Modelle können erheblich zur Verkürzung der Rechenzeit bei der Lösung hierarchischer Optimierungsprobleme beitragen. Zudem eignen sie sich für die Auslegung spezieller selbstoptimierender Regelungen. Beide Anwendungsgebiete werden in Kapitel 6 ausführlich behandelt. Die vorgestellte Methode unterliegt jedoch den folgenden drei Voraussetzungen:

- Bei den zu reduzierenden Systemen muss es sich um lineare Systeme mit beliebiger Parameterabhängigkeit handeln.
- Es erfolgt eine lineare Interpolation bzgl. der Parametrisierung α der paretooptimalen Systeme.
- Es können nur eindimensionale Paretomengen, d. h. Optimierungsprobleme mit zwei Zielfunktionen behandelt werden.

In den folgenden drei Abschnitten wird auf diese drei Beschränkungen eingegangen. Dabei wird ein Ausblick auf zukünftige Erweiterungsmöglichkeiten der Methode gegeben, indem erste Ideen für eine Aufhebung oder Abschwächung der Beschränkungen vorgestellt werden.

5.5.1 Behandlung nichtlinearer Systeme

Mechatronische und insbesondere selbstoptimierende Systeme zeigen sehr oft nichtlineares Systemverhalten. In vielen Fällen, wie beispielsweise bei dem in dieser Arbeit betrachteten Feder-Neige-Prüfstand, kann das Systemverhalten hinreichend genau unter Verwendung linearer Systeme beschrieben werden. Für andere Systeme wird dies nicht der Fall sein. Da eine Mehrzieloptimierung ohne Weiteres auch basierend auf einem nichtlinearen Optimierungsmodell durchgeführt werden kann, wäre eine nachfolgende MOR der nichtlinearen paretooptimalen Systeme wünschenswert und würde den Anwendungsbereich der in dieser Arbeit vorgestellten Methode deutlich ausweiten.

Unabhängig von der hier betrachteten, speziellen Aufgabenstellung der Reduktion paretooptimaler Systeme, stellt die MOR nichtlinearer Systeme selbst noch eine Herausforderung dar. Bestehende Verfahren wie die POD (Proper Orthogonal Decomposition, siehe [Ant05] für eine Einführung) oder auch die bereits in Abschnitt 3.2 erwähnte TPWL (Trajectory-Piecewise-Linear) benötigen feste Trajektorien oder Trainingsdaten, die zur Konstruktion des reduzierten Modells verwendet werden. Eine Berücksichtigung zusätzlicher Parameter wird hierdurch erschwert und wurde – soweit bekannt – noch nicht vorgenommen.

Ein anderer, sehr neuer Ansatz zur Modellordnungsreduktion nichtlinearer Systeme ist die Reduktion bilinearer Systeme. Dies sind Systeme der Form

$$\begin{aligned}\dot{x}(t) &= Ax(t) + \sum_{i=1}^m N_i x(t) u_i(t) + Bu(t), \\ y(t) &= Cx(t),\end{aligned}\tag{5-13}$$

die ebenfalls mit Hilfe von Krylov-Unterräumen reduziert werden können, siehe z.B. [BD10]. In [Gu11] wird darüber hinaus ein Verfahren beschrieben, mit dem eine Reihe von nichtlinearen Systemen als sogenannte *quadratische bilineare Systeme* ausgedrückt werden können. Im Unterschied zu bisherigen Arbeiten, die allgemeine nichtlineare Systeme beispielsweise mit Hilfe einer Taylor-Entwicklung durch bilineare Systeme approximieren, handelt es sich hierbei nicht um eine Approximation, sondern eine äquivalente Darstellung des Systems. Ein quadratisches bilineares System enthält zusätzlich zu (5-13) einen Ausdruck, der quadratisch von dem Zustand x abhängt. Das quadratische bilineare System ist dann gegeben als

$$\begin{aligned}\dot{x}(t) &= A_1 x(t) + A_2 (x(t) \otimes x(t)) + \sum_{i=1}^m N_i x(t) u_i(t) + Bu(t), \\ y(t) &= Cx(t),\end{aligned}$$

wobei \otimes das Tensorprodukt darstellt und dementsprechend $A_2 \in \mathbb{R}^{n_x \times n_x^2}$ gilt.

Der auch im Rahmen dieser Arbeit eingesetzte IRKA (Algorithmus 2 in Abschnitt 2.4) wurde in [BB12] sowohl auf bilineare, als auch auf quadratische bilineare Systeme erweitert. Darüber hinaus werden in [BB11] Systeme mit affin-linearen Parameterabhängigkeiten als bilineare Systeme formuliert und mit Hilfe des modifizierten IRKA reduziert. Über die Darstellung als bilineares System könnte sich daher zukünftig ein Ansatz ergeben, parametrische nichtlineare Systeme zu reduzieren.

5.5.2 Interpolation höherer Ordnung

Die Interpolation der paretooptimalen Systeme erfolgt in dieser Arbeit durch lineare, matrixwertige Splinefunktionen. Um eine bessere Approximation der paretooptimalen Systeme mit weniger Stützstellen zu erzielen, wäre eine Interpolation höherer Ordnung, z. B. mit Hilfe kubischer Splines, von Vorteil. Zudem wären die interpolierten Systeme nicht nur stetig, sondern (mehrfach) stetig differenzierbar. Dies könnte besonders bei der Anwendung der Methode auf mehreren Ebenen der Systemhierarchie von Vorteil sein. In diesem Fall werden die parametrischen reduzierten Systeme selbst als Teilsysteme der nächsthöheren Hierarchieebene für die Parametrisierung und Interpolation verwendet.

Eine komponentenweise Spline-Interpolation höherer Ordnung auf Basis der Parametrisierung α ist zunächst ohne weiteres möglich. Anstatt einer stückweise linearen, matrixwertigen Funktion, erhält man eine stückweise polynomielle Funktion

$$A(\alpha) = A_{j,0} + A_{j,1}\alpha + A_{j,2}\alpha^2 + \dots, \quad j = 1, \dots, n_S$$

der Dynamikmatrizen. Ein derartiger kubischer Spline ist beispielsweise die Funktion s_3 , die bereits bei der Bestimmung der Stützstellen für die Interpolation in Abschnitt 5.2 verwendet wurde.

Die manuelle rationale Interpolation kann auch bei polynomiellen statt affin-linearen Parameterabhängigkeiten genutzt werden. In [DSC⁺04] geschieht dies durch die Einführung zusätzlicher Parameter für die höheren Potenzen. Als alternativer Weg werden in [FHIDE06] spezielle Krylov-Unterräume konstruiert, sodass polynomielle Parameterabhängigkeiten berücksichtigt werden können. Da die manuelle rationale Interpolation allerdings nur lokal auf jeden Abschnitt einzeln angewendet werden kann, entstehen an den Stützstellen stets Unstetigkeiten bei der Reduktion sowohl in den Systemmatrizen selbst, als auch in den Ableitungen bzgl. α .

Dies ließe sich vermeiden, wenn die Spline Struktur auch in den reduzierten Systemen erhalten bleiben könnte, wie das bei der linearen Interpolation in Kombination mit der Matrix Interpolation möglich ist. Die Darstellung beliebiger Splines

als B-Splines weist gewisse Ähnlichkeiten mit der für die Matrix Interpolation geforderten Struktur (3-15) auf. Für eine allgemeine Einführung der B-Splines sei auf [Wer92] oder [Boo01] verwiesen. Bezeichnet man mit $B_{i,k}$ den B-Spline der Ordnung k zum Knotenpunkt i , so ist ein beliebiger Spline sp darstellbar als Summe

$$sp(\alpha) = \sum_{i=0}^{n_K} a_i B_{i,k},$$

wobei n_K die Anzahl der Knoten bezeichnet, die für Interpolationen höherer Ordnung nicht der Anzahl der zu interpolierenden Funktionswerte entspricht. Die Koeffizienten a_i können effizient durch die Lösung linearer Gleichungssysteme bestimmt werden.

Im Falle einer matrixwertigen Interpolation ergibt sich

$$A(\alpha) = \sum_{i=0}^{n_K} \mathcal{A}_i B_{i,k} \text{ mit } \mathcal{A}_i \in \mathbb{R}^{n_x \times n_x}.$$

Für die lineare Interpolation erhält man daraus das auch in dieser Arbeit genutzte Resultat

$$A(\alpha_i) = \mathcal{A}_i,$$

das eine separate Reduktion der Systeme an den Stützstellen α_i mit einer anschließenden Interpolation ermöglicht. Für Interpolationen höherer Ordnung gilt dieser Zusammenhang jedoch nicht mehr, da der Träger von $B_{i,k}$ immer aus $k+1$ Knotenpunkte besteht. Es gilt dann lediglich

$$A(\alpha_j) = \sum_{i=j}^{j+k} \mathcal{A}_i B_{i,k},$$

woraus sich ein Problem für die MOR ergibt. Die B-Splines selbst bilden zwar auch für höhere Ordnungen eine *Zerlegung der Eins*, d. h.

$$1 \equiv \sum_{i=0}^{n_K} B_{i,k}(\alpha), \quad \forall \alpha,$$

weshalb sie als Gewichtsfunktionen für die Matrix Interpolation geeignet wären. Allerdings besitzen die Koeffizienten \mathcal{A}_j keine systemtheoretische Bedeutung mehr. Eine MOR mit diesen Koeffizientenmatrizen als Dynamikmatrizen ist fraglich und bedarf weiteren Untersuchungen. Es ist zudem offen, ob und auf welche Weise sich die notwendigen Rückprojektionen auf einen gemeinsamen Unterraum für Interpolationen höherer Ordnung anwenden lassen.

5.5.3 Behandlung von drei oder mehr Zielfunktionen

Die Paretomenge eines Mehrzieloptimierungsproblems stellt unter gewissen Annahmen an die Zielfunktionen ein $(k - 1)$ -dimensionales Objekt dar, wobei k die Anzahl der Zielfunktionen angibt. Eine Parametrisierung muss aus einer der Dimension entsprechenden Anzahl von Parametern bestehen. Bei drei oder mehr Zielfunktionen ist daher zunächst eine geeignete Parametrisierung zu wählen.

Eine orthogonale Projektion auf einen Simplex, der von den Minima der einzelnen Zielfunktionen aufgespannt wird, ist nur für den Fall von zwei Zielfunktionen geeignet. Bei drei oder mehr Zielfunktionen können projizierte Punkte außerhalb des Simplex liegen, siehe [Hil01] für ein einfaches Beispiel. Eine Parametrisierung ist noch möglich, wenn statt des Simplex die gesamte Ebene verwendet wird, was lediglich zur Folge hat, dass die Parameterwerte außerhalb des Intervalls $[0, 1]$ liegen.

Die Paretopunkte können ebenfalls mit Hilfe von zwei Winkeln beschrieben werden. Kombinationen aus einem Zielfunktionswert und einem Winkel, der das Verhältnis der beiden anderen Zielfunktionen beschreibt, sind ebenfalls denkbar.

Eine Interpolation der paretooptimalen Systeme kann auch in Abhängigkeit von zwei (oder mehr) Parametern durchgeführt werden. Hierfür sind z.B. bilineare Tensorsplines geeignet wie sie in [SK11] eingeführt werden. Diese stellen ein Produkt aus eindimensionalen Interpolationen mit linearen B-Splines dar. Verwendet man nur lineare Interpolationen, so können sowohl die manuelle rationale Interpolation, als auch die Matrix Interpolation zur Reduktion verwendet werden. Die manuelle rationale Interpolation wird erneut für jedes Segment separat angewendet. Die Anzahl der zu berücksichtigenden Parameter entspricht der Dimension der Paretomenge. Bei der Matrix Interpolation werden reduzierte Systeme an den Stützstellen berechnet. Diese können dann wie im eindimensionalen Fall linear interpoliert werden.

Zu erweitern ist der Algorithmus zur Bestimmung der Anzahl und der Lage der Stützstellen. Die hier verwendete Funktion *newnot* kann nur auf Splines angewendet werden, die von einem einzigen Parameter abhängen. Für einen ersten Ansatz ist auch eine äquidistante Verteilung denkbar. Diese wird beispielsweise in [BDD12] für die Parametrisierung von Paretomengen mit drei Zielfunktionen verwendet. Die Grundlage der Parametrisierung ist dort eine Interpolation mit Hilfe von B-Splines.

6 Anwendungsgebiete parametrischer reduzierter Modelle

Dieses Kapitel zeigt Anwendungsmöglichkeiten der parametrischen reduzierten Systeme auf. Zum einen wird der Einsatz der reduzierten Modelle im Bereich der hierarchischen Optimierung illustriert. Als Anwendungsbeispiele dienen eine erneute Optimierung des Feder-Neige-Prüfstands mit reduzierten Modellen der beiden Aktormodule in Abschnitt 6.1 sowie die Optimierung eines vernetzten Prüfstands in Abschnitt 6.2. In diesem zweiten Anwendungsbeispiel wird das parametrische reduzierte Modell des gesamten Feder-Neige-Prüfstands eingesetzt.

Zum anderen wird in Abschnitt 6.3 mit der zielfunktionsbasierten Regelung ein Konzept zur Umsetzung der Selbstoptimierung kurz vorgestellt, in dem die parametrischen reduzierten Modelle ebenfalls eingesetzt werden können. Neben einer Darstellung der Einsatzmöglichkeiten der parametrischen reduzierten Modelle unterstreichen die in diesem Kapitel präsentierten Ergebnisse noch einmal die hohe Approximationsgüte, die mit der im Rahmen dieser Arbeit entwickelten Methodik erreicht werden kann.

6.1 Hierarchische Optimierung des Feder-Neige-Prüfstands

Die hierarchische Optimierung des Feder-Neige-Prüfstands basiert auf dem hierarchischen Modell, das in Abschnitt 1.3 vorgestellt wird. Bei der hierarchischen Optimierung werden voneinander abhängige, hierarchisch angeordnete Mehrzieloptimierungsprobleme (MOP) betrachtet, siehe Abschnitt 4.2 für eine ausführliche Darstellung.

Im Fall des Feder-Neige-Prüfstands bilden die beiden Aktormodule die untere Hierarchieebene. Eine Mehrzieloptimierung eines einzelnen Aktormoduls ist in Abschnitt 4.4.1 beschrieben. Die im Folgenden dargestellten Ergebnisse verwenden die dort gezeigte Paretofront bzw. Paretomenge.

Im Unterschied zu der bisher betrachteten Optimierung des Feder-Neige-Prüfstands (siehe Abschnitt 4.4.2) wird in diesem Abschnitt das hierarchische Modell im Optimierungsmodell genutzt. Die Aktormodule sind daher nun variabel konfigurierbar, wobei die zulässigen Konfigurationen auf die mittels der Optimierung bestimmten optimalen Systemeinstellungen beschränkt werden. Die Zielfunktionen sind identisch zu der bereits beschriebenen Optimierung und bewerten den Komfort sowie den Energieverbrauch des Systems, siehe (4-5). Die optimalen Systemkonfigurationen der Aktormodule sind durch die Parametrierung α gegeben.

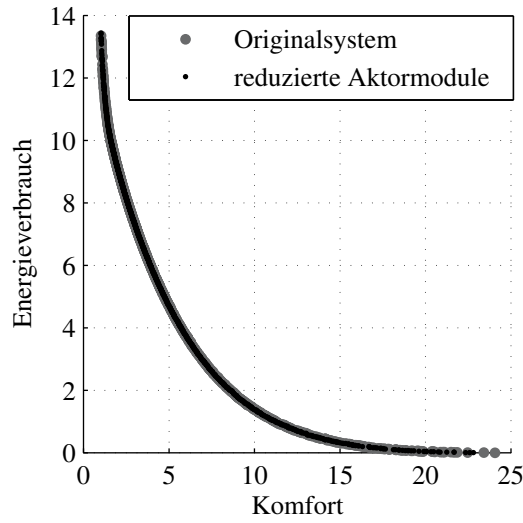


Bild 6-1: Vergleich der Paretofronten der hierarchischen Optimierung des Feder-Neige-Prüfstands mit reduzierten und nicht reduzierten Aktormodulen

Im Folgenden werden zur Vereinfachung die beiden Aktormodule durch dieselbe Variable parametrisiert, verhalten sich also stets identisch.

Wie in Abschnitt 4.2 dargestellt, wird α neben den Optimierungsparametern der oberen Ebene als zusätzliche Optimierungsvariable verwendet. Insgesamt ergibt sich hieraus für die obere Ebene das MOP

$$\min_{p, \alpha} \{ [f_1(p, \alpha), f_2(p, \alpha)] : p \in \mathcal{D} \subseteq \mathbb{R}^3, 0 \leq \alpha \leq \alpha_{max} \},$$

mit den Zielfunktionen f_1 und f_2 aus (4-5) und dem in Abschnitt 4.4.2 beschriebenen Optimierungsmodell.

Bei der Implementierung werden zur Umsetzung einer Funktionsauswertung zu einem gegebenen α die zugehörigen Zustandsdarstellungen der Aktormodule berechnet und in das hierarchische Modell eingebunden. An dieser Stelle kann das parametrische reduzierte Modell der Aktormodule aus Abschnitt 5.4.1 eingesetzt werden. Die parametrisch reduzierten Aktormodule hängen direkt von der Optimierungsvariable α ab und können daher bei der Implementierung besonders schnell und einfach variiert werden. Zudem sind keine Detailkenntnisse mehr über das System und seine ursprünglichen Parameter notwendig. Ein weiterer Vorteil der reduzierten Modelle liegt darin, dass die in jeder Funktionsauswertung ausgeführte Simulation des Optimierungsmodells durch die Verwendung eines reduzierten Modells beschleunigt wird.

In den Bildern 6-1 und 6-2 sind die Ergebnisse der hierarchischen Optimierung sowohl unter Verwendung der reduzierten, als auch der nicht reduzierten Aktor-

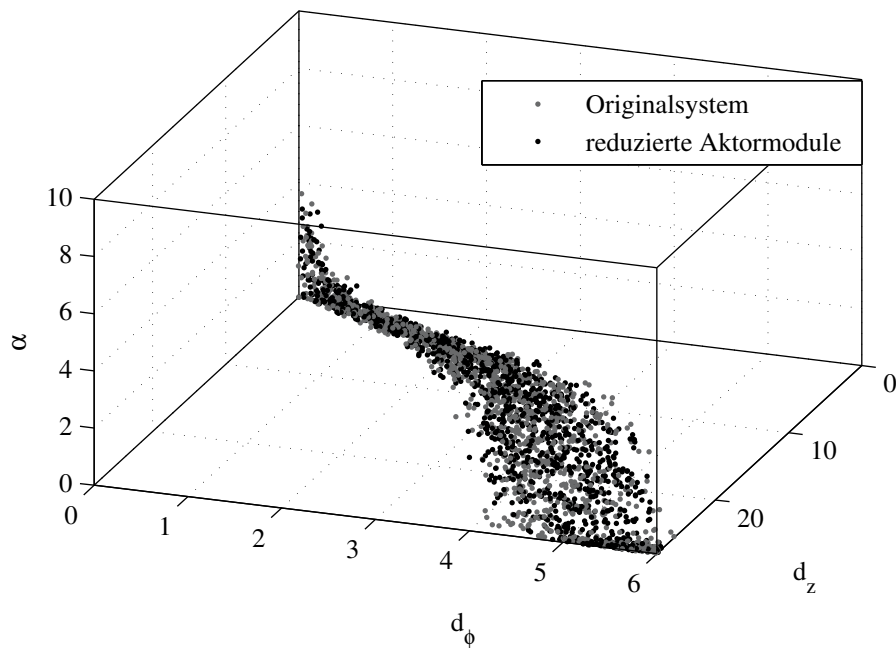


Bild 6-2: Vergleich der Paretomengen der hierarchischen Optimierung des Feder-Neige-Prüfstands mit reduzierten und nicht reduzierten Aktormodulen

module einander gegenüber gestellt. Die berechneten Paretofronten und Paretomengen stimmen sehr gut überein. Im Bildraum, der die Zielfunktionswerte in skaliert Form wiedergibt, ist optisch kaum ein Unterschied zwischen den beiden Paretofronten zu erkennen. Da für die Optimierung der *sampling*-Algorithmus benutzt wurde, sind im Urbildraum aufgrund der zufälligen Wahl der Testpunkte andere Paretopunkte berechnet worden. Die Paretomengen stimmen aber qualitativ sehr gut überein.

Ein Vergleich der Ausführungszeiten für eine Funktionsauswertung ist in Tabelle 6-1 dargestellt. Insgesamt kann die Rechenzeit für eine vollständige Funktionsauswertung durch das parametrische reduzierte Modell um ca. 33 % verringert

Tabelle 6-1: Vergleich der durchschnittlichen Ausführungszeiten

	original	reduziert	Ersparnis
Funktionsauswertung	4,64 s	3,098 s	33 %
Simulation	3,165 s	3,067 s	3 %
Initialisierung	1,475 s	0,031 s	98 %

werden. Auffällig ist hier allerdings, dass die eigentliche Simulation des Optimierungsmodells nur geringfügig, um 3%, schneller ausgeführt wird. Die hauptsächliche Ersparnis resultiert aus einer signifikanten Verkürzung der Initialisierung des Modells. Bei nicht reduzierten Aktormodulen wird zu einem gegebenem Wert von α zunächst die Paretomenge interpoliert, um die Systemparameter zu erhalten. Anschließend werden für das linke und das rechte Aktormodul jeweils ein Simulinkmodell linearisiert, um automatisiert eine Zustandsdarstellung der Aktormodule zu erhalten, die dann eingebunden werden kann. Dieses Vorgehen ist durchaus typisch, wenn die zu optimierenden Modelle unter Simulink erstellt wurden und eine analytische Beschreibung nicht vorliegt. Beide Schritte werden beim reduzierten System ersetzt durch eine einfache Auswertung der affinen Abhängigkeit der reduzierten Systeme von α , d. h. durch die Berechnung $A(\alpha) = A_{r,i} + \bar{\alpha}\Delta A_{r,i}$. Durch diese signifikante Vereinfachung der Initialisierung ist die erhebliche Verringerung der Rechenzeit zu erklären. Dies unterstreicht den Vorteil der gekapselten, abstrakten Form der reduzierten Systeme.

6.2 Hierarchische Optimierung eines vernetzten Prüfstands

In diesem Abschnitt dient ein weiteres hierarchisches Optimierungsproblem als Anwendungsbeispiel. Eingesetzt wird dabei das parametrische reduzierte System des gesamten Feder-Neige-Prüfstands, das in Abschnitt 5.4.2 ausführlich beschrieben ist. Optimiert wird ein vernetzter Prüfstand, der neben dem Feder-Neige-Prüfstand aus einem weiteren Prüfstand aus dem Kontext des RailCab, dem intelligenten Antriebsmodul, besteht.

Im folgenden Abschnitt 6.2.1 wird zunächst der Prüfstand des intelligenten Antriebsmoduls beschrieben. Anschließend wird in Abschnitt 6.2.2 das hierarchische Modell des vernetzten Prüfstands vorgestellt. In Abschnitt 6.2.3 wird dann auf die hierarchische Optimierung eingegangen. Der Schwerpunkt liegt dabei auf dem Vergleich zwischen der Optimierung mit nicht reduziertem und mit reduziertem Modell des Feder-Neige Prüfstands.

6.2.1 Intelligentes Antriebsmodul

Das RailCab wird durch einen doppelt gespeisten Linearmotor angetrieben, wie in Abschnitt 1.3 bereits kurz erwähnt. Dieser wird durch einen in der Gleismitte fixierten Stator und in das RailCab eingebaute Läufer realisiert. Die Kraftübertragung erfolgt berührungslos, sodass zwischen Stator und Läufer ein Luftspalt vorhanden ist. Die Größe dieses Luftspalts hat einen erheblichen Einfluss auf die elektrischen Verluste und somit auf den Wirkungsgrad des RailCabs. Dabei führt ein geringerer Luftspalt zu geringeren Verlusten.

Aufgrund baulicher Ungenauigkeiten, durch Setzungs- und Alterungsprozesse sowie Radverschleiß variiert der Luftspalt während der Fahrt in Abhängigkeit von

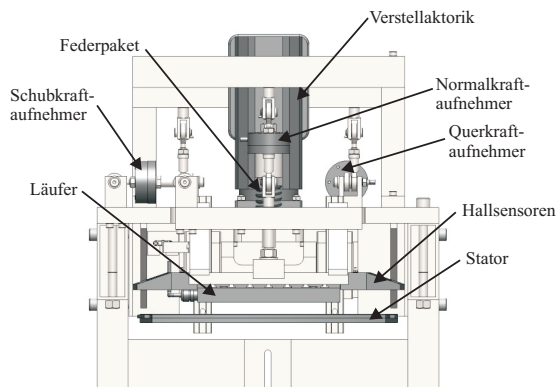
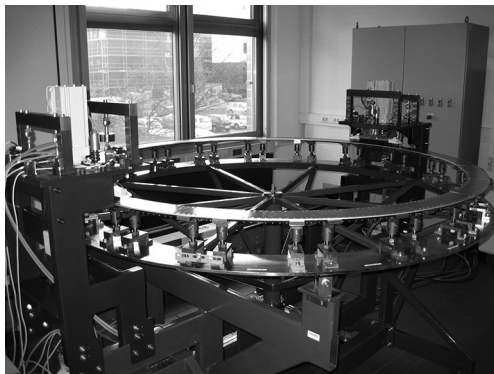


Bild 6-3: links: Prüfstand für das intelligente Antriebsmodul; rechts: Aktormodul für die aktive Luftspaltverstellung [HHKS08]

der momentan befahrenen Strecke. Aus diesem Grund ist ein intelligentes Antriebsmodul entwickelt worden [SZ05, HHKS08], mit dem eine aktive Verstellung des Luftspalts möglich ist. Auf diese Weise ist es möglich, die Energieverluste zu verringern. In diesem Abschnitt wird der Prüfstand dieses intelligenten Antriebsmoduls betrachtet.

Der Prüfstand, abgebildet in Bild 6-3 links, besteht aus einem Rondell, das den Stator repräsentiert, und zwei Aktormodulen, die jeweils einen Läufer des Rail-Cabs mit aktiver Luftspaltverstellung nachbilden. Durch die Realisierung des Stators als geschlossener Kreis ist eine Untersuchung beliebig langer Streckenabschnitte möglich. Die Fahrgeschwindigkeit des Fahrzeugs kann durch Variationen der Drehgeschwindigkeit des Rondells berücksichtigt werden. Der Stator stellt im Gegensatz zum RailCab aus Kostengründen keinen doppelt gespeisten Asynchronlinearmotor, sondern einen Permanentmagnet erregten Synchronlinearmotor dar. Die vertikale Position der einzelnen Permanentmagnete kann manuell verändert werden. Auf diese Weise können unterschiedliche Störungen der Strecke nachgebildet werden.

Der Aufbau der beiden Aktormodule ist in Bild 6-3 rechts dargestellt. Sie bestehen aus einem Läufer, der Verstellaktorik und einer Reihe von Kraftsensoren (Schubkraft-, Normalkraft- und Querkraftaufnehmer). Zudem wird der Läufer von einer passiven Feder gehalten, die bewirkt, dass die Verstellaktorik nicht die gesamten auftretenden Normalkräfte zwischen Läufer und Stator aufbringen muss und somit die elektrischen Verluste verringert werden. Des Weiteren stellen die Federn eine mechanische Notfallebene dar, um bei einem Ausfall des Verstellaktors den Linearmotorläufer in seine Endlage zu ziehen. Hallsensoren sind am Läufer befestigt und werden u. a. dazu genutzt, den momentanen Luftspalt sowie die Rondellgeschwindigkeit zu messen.

In [HKK⁺12] ist ein nichtlineares Modell des Prüfstands beschrieben, das auch in dieser Arbeit verwendet wird. Es bildet das dynamische Verhalten hinreichend genau nach. In dem Modell enthalten sind u. a. die nichtlinearen Charakteristiken der Magnetkräfte. Einflüsse von Reibung und Luftwiderstand werden ebenfalls simuliert.

Auf Basis des nichtlinearen Modells wird ein Optimierungsmodell für das intelligente Antriebsmodul erstellt und eine Mehrzieloptimierung durchgeführt, siehe erneut [HKK⁺12]. Für die Mehrzieloptimierung des Prüfstands werden zwei Ziele definiert. Zum einen sind dies die Energieverluste der Verstellaktorik und zum anderen die Energieverluste des Linearantriebs. Diese beiden Ziele sind gegenläufig. Bei langsamen Geschwindigkeiten wird wenig Energie für den Linearantrieb aufgewendet. Die benötigte Energie für die Verstellaktorik steigt jedoch, da ein kleinerer Luftspalt realisiert werden kann. Umgekehrt vergrößert sich aus Sicherheitsgründen der Luftspalt bei hohen Geschwindigkeiten. Dies hat zur Folge, dass weniger Energie für die Verstellaktorik aber aufgrund der gestiegenen Geschwindigkeit mehr Energie für den Linearantrieb aufgebracht werden muss.

Es werden drei Optimierungsparameter gewählt. Dies sind zum einen Verstärkungsfaktoren $K_{p,up}$ und $K_{p,down}$ aus der Regelung der Verstellaktorik. Sie beeinflussen vor allem die Schnelligkeit, mit der die Verstellaktorik auf einen wechselnden Luftspalt reagiert. Dabei wird zwischen einer Aufwärts- und einer Abwärtsbewegung (up/down) unterschieden. Als dritter Optimierungsparameter wird die Geschwindigkeit v_{track} des Rondells betrachtet.

In Bild 6-4 sind die Ergebnisse der Mehrzieloptimierung dargestellt. Im linken Teil ist die Paretofront, im rechten Teil die zugehörige Paretomenge zu finden. Für die Mehrzieloptimierung wurde erneut der *sampling*-Algorithmus eingesetzt, siehe Kapitel 4. Die hier dargestellte Paretomenge wird als Grundlage für die hierarchische Optimierung des im nächsten Abschnitt beschriebenen vernetzten Prüfstands genutzt. Eine weitergehende Analyse der Optimierungsergebnisse ist in [HKK⁺12] zu finden.

6.2.2 Vernetzter Prüfstand

Die beiden Prüfstände, der Feder-Neige-Prüfstand und das intelligente Antriebsmodul, sind bisher getrennt voneinander betrachtet worden. Eine Mehrzieloptimierung bzgl. der jeweiligen Ziele ist im vorhergehenden Abschnitt für das intelligente Antriebsmodul und in Abschnitt 4.4 für den Feder-Neige-Prüfstand beschrieben worden.

In der Realität, d. h. im RailCab, werden beide Systeme zusammen eingesetzt, wodurch Wechselwirkungen auftreten. Im Fall dieser beiden Systeme resultieren die Wechselwirkungen vor allem aus Geschwindigkeitsänderungen. Diese verändern zum einen die Frequenz, mit der Störungen auftreten, die von der aktiven

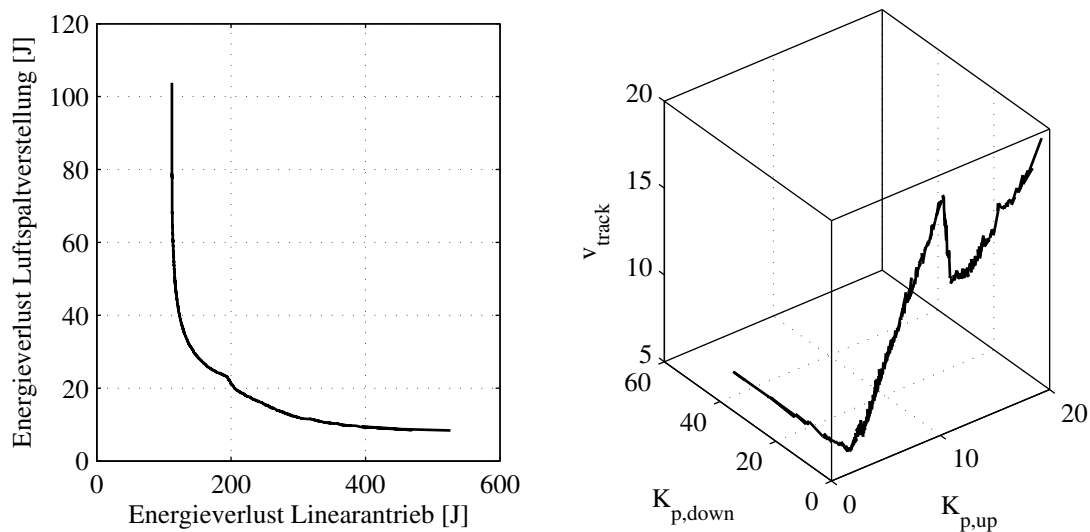


Bild 6-4: Paretofront (links) und Paretomenge (rechts) des intelligenten Antriebsmoduls

Federung gedämpft werden. Zum anderen verändern sich die Energieverluste für Antrieb und Aktorik der Luftspaltverstellung bei wechselnden Geschwindigkeiten.

Um diese Wechselwirkungen bereits im Vorfeld untersuchen zu können, ist die Idee eines vernetzten Prüfstands entstanden. Als eine Art Vorstufe von Tests am gesamten RailCab werden die Prüfstände so miteinander verkoppelt, dass zumindest die wesentlichen gegenseitigen Einflüsse untersucht werden können.

Bei den beiden Prüfständen handelt es sich um autonome mechatronische Systeme (AMS), sodass eine Kopplung nur über die Informationsverarbeitung erfolgen kann. Das resultierende Gesamtsystem, der vernetzte Prüfstand, ist dann selbst ein vernetztes mechatronisches System (VMS). Im ersten Schritt der Vernetzung wird ein gemeinsames Simulationsmodell erstellt, mit dem die Wechselwirkungen nachgebildet werden können. Die Kopplung wird hier konkret über ein gemeinsames Streckenprofil realisiert. Dazu werden für einen fiktiven Streckenabschnitt Störungen durch Gleislagefehler in vertikaler und horizontaler Richtung sowie Abweichungen in der vertikalen Position der Statoren definiert. Die Geschwindigkeit, mit der das Streckenprofil für die Simulation abgetastet wird, ist variabel. Sie stellt, wie bereits erwähnt, einen Parameter des intelligenten Antriebsmoduls dar. Das Simulationsmodell ist ein hierarchisches Modell, das in Bild 6-5 angedeutet ist.

Auf Basis der Optimierungsergebnisse der beiden einzelnen Module wird eine hierarchische Optimierung des vernetzten Systems durchgeführt. Dazu werden

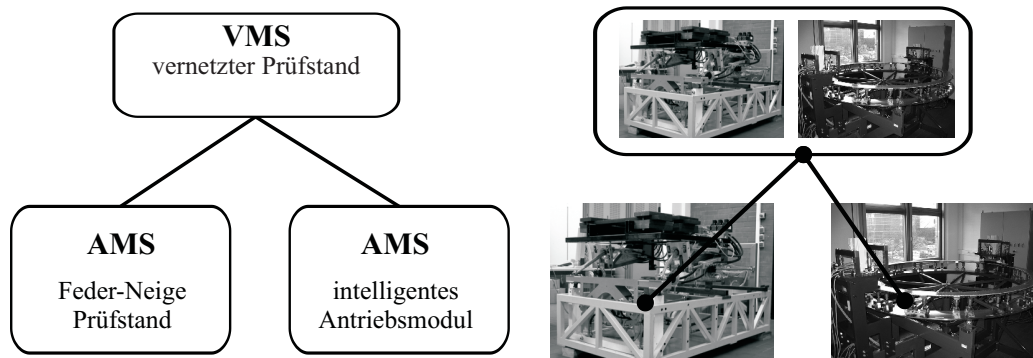


Bild 6-5: Hierarchische Zerlegung des vernetzten Prüfstands mit Angabe der Strukturelemente

die Paretomengen jeweils mit einem skalaren Parameter parametrisiert. Verwendet wird für beide Systeme der Simplex der Minima der einzelnen Zielfunktionen, siehe (5-3), mit $\alpha_{max} = 1$. Im gemeinsamen Simulationsmodell kann für den Feder-Neige-Prüfstand das parametrische reduzierte System aus Abschnitt 5.4.2 genutzt werden. Bei dem Modell des intelligenten Antriebsmoduls handelt es sich um ein nichtlineares Modell, in dem eine Reihe von Kennlinien enthalten sind. Aus diesem Grund lässt sich die in dieser Arbeit entwickelte PMOR nicht auf das intelligente Antriebsmodul anwenden. Im folgenden Abschnitt werden die Ergebnisse der hierarchischen Optimierung mit reduziertem und nicht reduziertem Modell des Feder-Neige-Prüfstands einander gegenüber gestellt.

6.2.3 Optimierungsergebnisse

Für den vernetzten Prüfstand wird in dieser Arbeit ein Mehrzieloptimierungsproblem mit zwei Zielfunktionen betrachtet. Die Zielfunktionen stellen eine Kombination der Zielfunktionen der unterlagerten Systeme dar. Dies ist eine von vielen denkbaren Möglichkeiten zur Definition von Zielfunktionen, beispielsweise wird in [HKK⁺12] eine hierarchische Optimierung dieses vernetzten Prüfstands mit drei Zielfunktionen durchgeführt. Eine PMOR wird dort allerdings nicht eingesetzt.

Die erste Zielfunktion besteht aus dem gesamten Energieverbrauch, der an den beiden Prüfständen vorliegt. Beim intelligenten Antriebsmodul wird die gesamte Leistung, d. h. die Summe aus Leistung für die Verstellaktorik und den Linearantrieb, gebildet und integriert. Die Leistung des Feder-Neige-Prüfstands wird wie bisher über die Sollkräfte der aktiven Federung angenähert. Beide Anteile sind für die Optimierung so gewichtet, dass sie in der gleichen Größenordnung liegen.

Als zweite Zielfunktion wird eine gewichtete Summe aus Fahrzeit für den Streckenabschnitt und gemessenem Komfort definiert. Im Folgenden wird diese Ziel-

funktion als Zeit/Komfort-Mix bezeichnet. Die Gewichte sind so gewählt, dass beide Anteile, die Fahrzeit und der Komfort, in etwa gleichwertig in die Zielfunktion eingehen. Der Komfort basiert dabei wie bei der Betrachtung des Feder-Neige-Prüfstands auf einem frequenzgewichteten und skalierten Mittelwert der Aufbaubeschleunigungen.

Die Optimierungsparameter des vernetzten Prüfstands bestehen nur aus den beiden Parametern α_{iAM} und α_{FNP} , die aus der Parametrierung der jeweiligen Parametromengen der Prüfstände stammen.

Das Mehrzielloptimierungsproblem wird erneut mit dem *sampling*-Algorithmus gelöst. In Bild 6-6 sind die Optimierungsergebnisse sowohl für das System mit reduziertem Feder-Neige-Prüfstand als auch für das Originalsystem mit nicht reduziertem Feder-Neige-Prüfstand dargestellt.

Im Bildraum ergibt sich eine Paretofront mit der üblichen hyperbelartigen Form. Die Angabe der Zielfunktionswerte wird an dieser Stelle vernachlässigt, da sie durch die Skalierung keine physikalische Bedeutung mehr besitzen und im Weiteren keine Rolle spielen. Im Urbildraum erhält man eine eindimensionale Paretomenge. Diese beginnt im Punkt $[1, 0]$ für minimalen Zeit/Komfort-Mix und endet im Punkt $[0, 1]$ für minimalen Energieverbrauch. Der Punkt $[1, 0]$ entspricht hierbei der komfortoptimalen Parametrierung des Feder-Neige-Prüfstands zusammen mit der Einstellung für minimale Verluste für die Luftspaltverstellung. Da dieser Punkt auch der maximalen Geschwindigkeit entspricht, ist eine Minimierung der Zeit/Komfort Zielfunktion plausibel. Am anderen Ende der Paretomenge im Punkt $[0, 1]$ wird der energieoptimale Punkt des Feder-Neige-Prüfstands zusammen mit dem minimalen Verlust aus dem Linearantrieb verwendet, der auch der geringsten Geschwindigkeit entspricht.

Die Ergebnisse für das reduzierte und das nicht reduzierte Modell des Feder-Neige-Prüfstands stimmen sehr gut überein. Im Bildraum sind Unterschiede in der gewählten Darstellung kaum erkennbar. Im Urbildraum gibt es kleinere Abweichungen. Die Charakteristik der Paretomenge sowie die Anfangs- und Endpunkte gleichen einander. Ein Einsatz des parametrischen reduzierten Systems ist ohne weiteres möglich. Die Zeitersparnis für eine Funktionsauswertung ist für den vernetzten Prüfstand jedoch minimal und beträgt lediglich 3.5 %. Dies liegt daran, dass das Modell des intelligenten Antriebsmoduls wesentlich aufwändiger zu simulieren ist und der Feder-Neige-Prüfstand im Vergleich dazu kaum ins Gewicht fällt. Da für den Feder-Neige-Prüfstand im Rahmen dieser Arbeit zudem eine analytische Darstellung der Systemmatrizen erarbeitet wurde, liegt im Gegensatz zu dem Modell aus dem vorhergehenden Abschnitt 6.1 auch keine Zeitersparnis bei der Initialisierung vor.

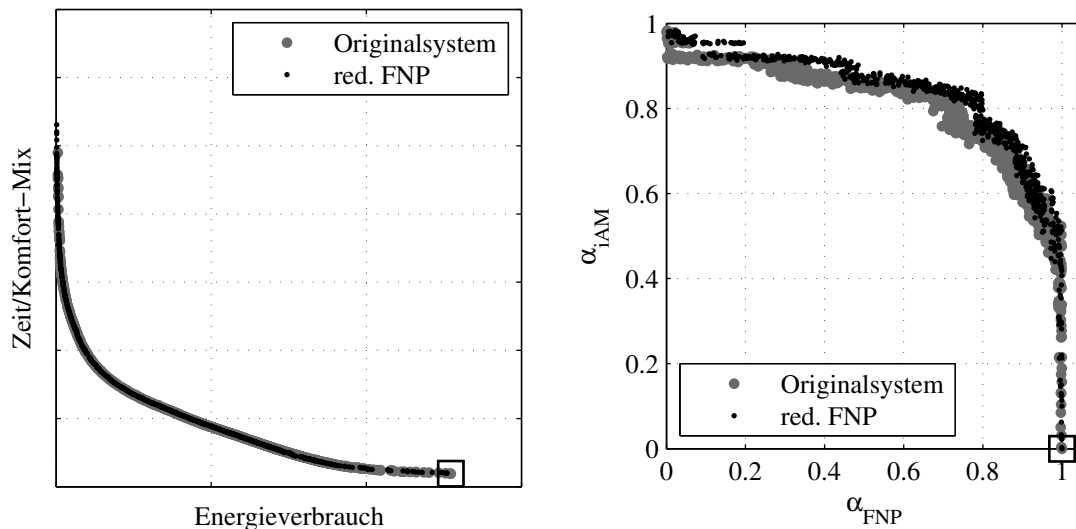


Bild 6-6: Paretofront (links) und Paretomenge (rechts) der hierarchischen Optimierung des vernetzten Prüfstands

6.3 Zielfunktionsbasierte Regelung selbstoptimierender Systeme

Die zielfunktionsbasierte Regelung adressiert die Umsetzung der Selbstoptimierung, nachdem eine Mehrzieloptimierung durchgeführt wurde. Während des Betriebs muss ein Punkt auf der berechneten Paretomenge ausgewählt werden. Dieser Vorgang, auch bezeichnet als *decision making*, stellt eine eigene Herausforderung dar. In [Mie99] sind Hinweise und weiterführende Literatur zu diesem Themengebiet zu finden.

In bisherigen Arbeiten zu selbstoptimierenden Systemen wird zum Teil der Ansatz verfolgt, Daten über zukünftige Situationen des Systems zu sammeln und für eine Planung zu nutzen. Das iterative Lernen von Störgrößenprofilen im RailCab-Kontext und eine hierauf basierende hybride Planung stellt ein Beispiel hierfür dar. Bei der hybriden Planung werden Simulationen mit paretooptimalen Konfigurationen für zukünftige Situationen, z. B. zukünftige Streckenabschnitte, durchgeführt. Auf Basis der Simulationsdaten wird dann die am besten passende Konfiguration ausgewählt, die z. B. vorgegebene Beschränkungen einhält, siehe [MAK⁺08] oder [AEH⁺11] für weitere Details. Ein wesentlicher Nachteil der hybriden Planung besteht darin, dass die zukünftigen Situationen des Systems, d. h. auch wesentliche Störgrößen und deren Verlauf, hinreichend genau bekannt sein müssen.

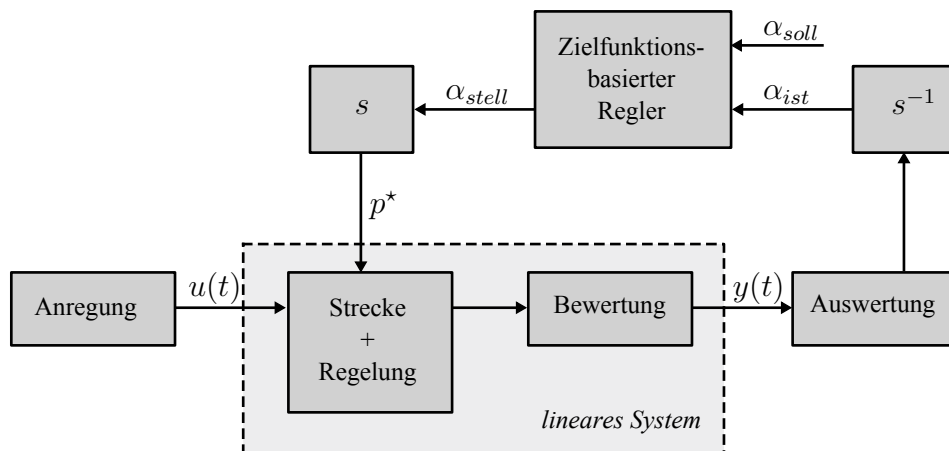


Bild 6-7: Struktur der zielfunktionsbasierten Regelung

In [GT09] wird als Alternative die Optimierung situationsabhängig zur Laufzeit durchgeführt. Allerdings liegen auch hier gewisse Beschränkungen vor. So muss das System über linear-quadratische Regelungen verfügen und der zu verwendende Paretopunkt muss darüber hinaus durch die Gewichte einer gewichteten Summe vorgegeben werden. Alternative Parametrierungen, wie sie in dieser Arbeit verwendet werden, sind nicht möglich. Der Ansatz besitzt allerdings den Vorteil, dass auch mehr als zwei Zielfunktionen verwendet werden können.

Der im Weiteren genauer vorgestellte Ansatz der zielfunktionsbasierten Regelung weist gewisse Ähnlichkeiten zu [GT09] auf in der Hinsicht, dass ebenfalls online bei unbekannten Störungen optimale Systemkonfigurationen autonom vom System gewählt werden. Die Grundidee geht auf [Mün12] zurück. Die hier präsentierte Form basiert auf [KRKT13], wo auch eine praktische Realisierung vorgestellt wird.

An Stelle einer Optimierung, die online durchgeführt wird, werden (wie bei der hybriden Planung) die offline berechneten Paretomengen genutzt. Hierdurch ist der Ansatz flexibler in Bezug auf die Zielfunktionen und die implementierte Regelung des Systems. Die Punktauswahl zur Laufzeit geschieht über eine spezielle zusätzliche Regelung. Hierbei handelt es sich um einen überlagerten Regelkreis, dessen Regelgröße die Parametrierung der Paretomenge darstellt. Zur Auslegung dieses überlagerten Reglers kann ein parametrisches reduziertes Modell eingesetzt werden, wodurch die Analyse der Regelung erheblich beschleunigt werden kann. Hierauf liegt im Folgenden der Schwerpunkt der Ausführungen.

Die Struktur der zielfunktionsbasierten Regelung ist in Bild 6-7 dargestellt. Sie ist eng verbunden mit der Struktur des Optimierungsmodells, siehe Bild 5-1, das

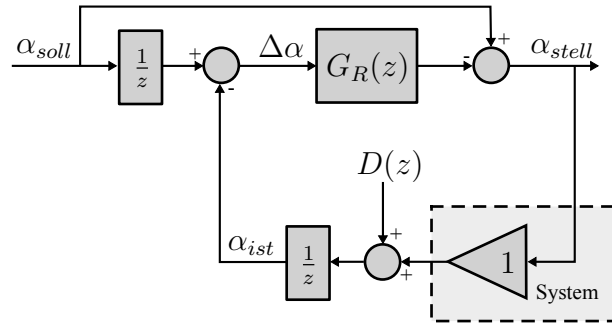


Bild 6-8: Abstraktes Modell für den Entwurf der zielfunktionsbasierten Regelung

für die Mehrzieloptimierung verwendet wird. Das grundsätzliche Ziel der überlagerten Regelung besteht darin, einen vorgegebenen Wert α_{soll} der Parametrierung der Paretomenge einzuregeln. Dieser repräsentiert beispielsweise ein gewisses Verhältnis der Zielfunktionen zueinander, wenn der Winkel des Zielfunktionsvektors als Parametrierung genutzt wird.

Die momentanen Zielfunktionswerte des Systems werden anhand der Messgrößen bestimmt. Mit Hilfe der Parametrierung, gegeben durch die Funktion $s : [0, \alpha_{max}] \rightarrow \mathbb{R}^{n_o}$ (vgl. Abschnitt 5.1), kann hieraus der aktuelle Wert α_{ist} berechnet werden. Der zielfunktionsbasierte Regler bestimmt auf Grundlage dieser beiden Größen die neue Stellgröße α_{stell} , die erneut durch die Parametrierung s in die zugehörige optimale Systemkonfiguration p^* umgerechnet wird. Diese wird von nun an vom System verwendet, wodurch der Regelkreis geschlossen ist.

In der hier dargestellten Form handelt es sich bei der zielfunktionsbasierten Regelung um eine diskrete Regelung mit einer relativ hohen Abtastzeit. Dies ist aufgrund der Auswertung der Zielfunktionen erforderlich. Da zumeist Durchschnittswerte bestimmter Größen berechnet werden, ist nach einer Anpassung der Systemkonfigurationen eine gewisse Zeit erforderlich bis die Zielfunktionen „eingeschwungen“ sind.

Zur Auslegung der zielfunktionsbasierten Regelung kann ein sehr einfaches, abstraktes Modell verwendet werden, das in Bild 6-8 dargestellt ist. Grundlage dieses abstrakten Modells ist die Annahme, dass eine Änderung der unterlagerten Regelung ohne Verzögerungen, wie z. B. Umschaltvorgänge, vorgenommen werden kann. Ist dies der Fall, besitzt das System keine Dynamik bezogen auf die Regelgröße α . Änderungen von α machen sich augenblicklich am Systemausgang y bemerkbar.

Die einzige Dynamik der abstrakten Regelstrecke resultiert aus der Auswertung der Zielfunktionen, die eine gewisse Zeit, nämlich genau eine Abtastperiode, benötigt. Im diskreten Bereich lässt sich dies durch die Übertragungsfunktion $\frac{1}{z}$ be-

rücksichtigen. Darüber hinaus wirken nicht näher bekannte Störungen $D(z)$ auf die Werte der Zielfunktionen ein. Sie resultieren aus den im Betrieb unbekannten Anregungen $u(t)$ auf das System. Hierbei ist für die Größe der Störung bzgl. α die Abweichung von der für die Optimierung verwendeten Anregung entscheidend. Sollte während des Betriebs genau die Störung aus der Mehrzieloptimierung auftreten, wäre $D(z) = 0$.

Als Regelung wird ein klassischer Eingrößenregler mit der diskreten Übertragungsfunktion $G_R(z)$ ergänzt um eine Vorsteuerung des gewünschten Wertes α_{soll} gewählt. Die Störübertragungsfunktion lässt sich leicht berechnen als

$$\frac{\Delta\alpha(z)}{D(z)} = \frac{-1}{z - G_c}.$$

Hieraus folgt mit Hilfe des Endwertsatzes, dass ein PI-Regler ausreicht, um konstante Störungen stationär genau auszuregeln. Daher wird der diskrete PI Regler

$$G_R = \frac{K_p(z - 1) + TK_i z}{z - 1}$$

mit den beiden Verstärkungen K_p und K_i gewählt. Die Funktionsfähigkeit dieses Reglers wurde bereits in [KRKT13] für den Feder-Neige-Prüfstand gezeigt. Daher wird im Folgenden nur auf die Verwendung des parametrischen reduzierten Modells eingegangen.

Für die Analyse der zielfunktionsbasierten Regelung sind Simulationen des Gesamtsystems erforderlich. Das System muss dabei aufgrund der großen Abtastzeit der überlagerten Regelung über einen längeren Zeitraum simuliert werden. Dies erfordert für komplexe Systeme wie den Feder-Neige-Prüfstand zeitintensive Berechnungen. Ein reduziertes System an Stelle des komplexen Gesamtsystems zu verwenden, kann die Rechenzeit signifikant verringern. Zu berücksichtigen ist dabei, dass sich das System in Abhängigkeit von α_{stell} in jedem diskreten Zeitschritt des zielfunktionsbasierten Reglers ändert. Da die mit der Methodik dieser Arbeit erstellten reduzierten Modelle explizit von der Parametrierung α abhängen, sind sie besonders geeignet für die Analyse der zielfunktionsbasierten Regelung.

Die Ein- und Ausgangsgrößen $u(t)$ bzw. $y(t)$ des unterlagerten Systems gleichen denen des Optimierungsmodells. Im Fall des Feder-Neige-Prüfstands wurden diese auch für das reduzierte Modell gewählt. Daher kann das parametrische reduzierte Modell ohne weitere Anpassungen auch für die Untersuchung der zielfunktionsbasierten Regelung verwendet werden. Nachfolgend wird ein Vergleich zwischen reduziertem Modell und nicht reduziertem Modell vorgenommen.

Für die zielfunktionsbasierte Regelung wird die in Kapitel 4 beschriebene Paretomenge verwendet. Als Parametrierung α wird der Winkel des Zielfunktionsvektors, siehe (5-2), mit $\alpha_{max} = 1000$ genutzt. Die Verstärkungsfaktoren des

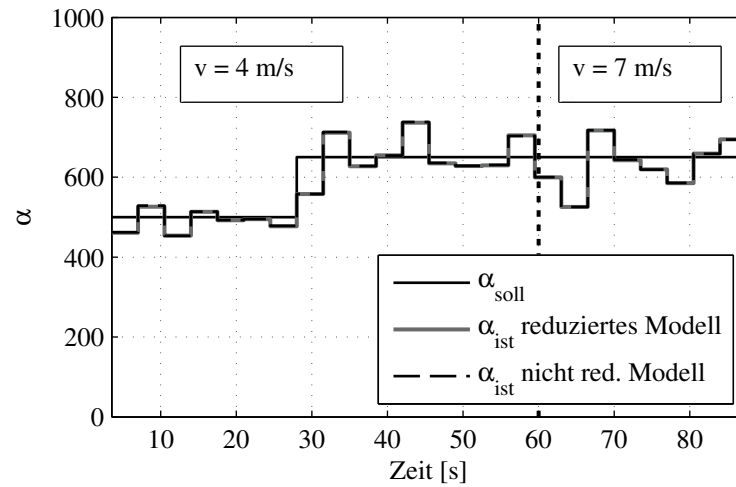


Bild 6-9: Vergleich des reduzierten mit dem nicht reduzierten Modell des Feder-Neige-Prüfstands

PI-Regler sind auf $K_p = 0.3$ und $K_i = 0.1$ gesetzt. Simuliert wird mit einer moderaten Anregung, die ein für den Prüfstand realistisches Streckenprofil darstellt. Das Profil wird in den ersten 60s der Simulation mit einer „virtuellen Geschwindigkeit“ von 4 m/s abgetastet. Anschließend wird die Geschwindigkeit auf 7 m/s angehoben, was zu einer Änderung der Anregungsfrequenz für den Prüfstand führt. Die Abtastrate für die überlagerte zielfunktionsbasierte Regelung beträgt $T = 3.5\text{ s}$.

Die Simulationsergebnisse für das vollständige lineare Modell sowie für das parametrisch reduzierte Modell, das in Abschnitt 5.4.2 beschrieben ist, sind in Bild 6-9 zu finden. Neben der bereits beschriebenen Geschwindigkeitsänderung wird nach 28 s der Wert von α_{soll} von 500 auf 650 angehoben. Die Funktionsfähigkeit des zielfunktionsbasierten Reglers ist deutlich zu erkennen. Beide Sollwerte werden gut eingeregelt. Die vorhandenen Abweichungen sind auf die stetig variierende Anregung zurückzuführen. Das reduzierte Modell stimmt zudem sehr gut mit dem nicht reduzierten System überein. In der gewählten Darstellung sind kaum Unterschiede erkennbar. Dies bestätigt, dass das parametrische reduzierte Modell an Stelle des Originalmodells verwendet werden kann. Ein Vergleich der reinen Simulationszeiten des reduzierten mit dem nicht reduzierten Modell für das konkrete Anregungsprofil zeigt, dass die Simulationszeit mit dem reduzierten Modell signifikant um ca. 93% gesenkt werden konnte. Diese hohe Verringerung ist darauf zurück zu führen, dass im Gegensatz zu den vorhergehenden Anwendungsfällen in diesem Fall nicht ein hierarchisches Modell mit dem reduzierten System auf der unterlagerten Ebene simuliert wird, sondern der Feder-Neige-Prüfstand das Gesamtsystem darstellt. Auf diese Weise wird die durch die Modellordnungsreduktion erzielbare Beschleunigung von Simulationen besonders deutlich.

7 Resümee und Perspektiven

In dieser Arbeit wird eine neuartige Methode zur parametrischen Modellordnungsreduktion paretooptimaler Systeme vorgestellt. Mit dieser Methode können die komplexen Modelle, die im Rahmen selbstoptimierender Systeme auftreten, gezielt vereinfacht werden. Das herausragende Merkmal der entwickelten Methodik besteht in der engen Verzahnung der Verfahren der parametrischen Modellordnungsreduktion mit der hierarchischen Optimierung auf der einen Seite und dem Konzept der hierarchischen Strukturierung und Modellierung mechatronischer Systeme auf der anderen Seite.

Es werden zwei Varianten der parametrischen Modellordnungsreduktion betrachtet. Die manuelle rationale Interpolation, angewendet durch einen im Rahmen dieser Arbeit entstandenen objektorientierten Arnoldi-Algorithmus einerseits und die Matrix Interpolation in Kombination mit der \mathcal{H}_2 -optimalen tangentialen Interpolation andererseits.

Beide Ansätze zur Modellordnungsreduktion profitieren erheblich von der neu entstandenen Methode zur Interpolation paretooptimaler Systeme. Mit Hilfe geeigneter Parametrierungen und einer automatisiert durchführbaren Diskretisierung der Paretomenge werden zum einen die optimalen Systemkonfigurationen zusammen mit dem zugehörigen dynamischen System zu einer Einheit gekapselt. Zum anderen verringert sich die Komplexität der parametrischen Reduktion, da die Anzahl beizubehaltender Parameter nur noch von der Anzahl der Zielfunktionen abhängt. Insbesondere durch die Kombination der Interpolation paretooptimaler Systeme zusammen mit der Matrix Interpolation ist eine weitgehend automatisierte Erstellung parametrischer reduzierter Systeme möglich.

Am Beispiel des Feder-Neige-Prüfstands, einem Prüfstand für die aktive Federung des Schienenverkehrssystems RailCab, kann die hohe Approximationsgüte der Reduktion nachgewiesen werden. Anhand von zwei Anwendungsfällen der hierarchischen Optimierung wird der Einsatz der parametrischen Modelle demonstriert. Darüber hinaus ist mit der zielfunktionsbasierten Regelung selbstoptimierender Systeme ein weiteres Anwendungsgebiet der parametrischen reduzierten Modelle identifiziert und erste Untersuchungen erfolgreich durchgeführt worden.

Die parametrische Modellordnungsreduktion paretooptimaler Systeme unterliegt in der hier vorgestellten Form einer Reihe von Voraussetzungen. Auf einige Erweiterungsmöglichkeiten ist bereits ausführlich in Abschnitt 5.5 eingegangen worden. Die dort geschilderten Ansätze zielen darauf ab, die Methode grundlegend zu erweitern und den Anwendungsbereich zu vergrößern. Darüber hinaus gibt es noch

eine Reihe von Aspekten, die zur weiteren Verbesserung der vorgestellten Methodik zukünftig angegangen werden können.

Als Anwendungsbeispiele dienen in dieser Arbeit ausschließlich hierarchische Systeme mit zwei Hierarchieebenen. Die Anwendbarkeit der Methode auf Systeme mit mehr als zwei Hierarchieebenen ist prinzipiell gegeben, aber in Zukunft noch zu validieren. An dieser Stelle ist vor allem der Einfluss der Unstetigkeiten der parametrischen reduzierten Modelle bzgl. des Reduktionsparameters zu untersuchen und ggf. durch geeignete Mittel zu verringern.

Die Implementierung des parametrischen reduzierten Modells, insbesondere im Kontext der zielfunktionsbasierten Regelung ist momentan relativ aufwändig. Hier könnte zukünftig etwa spezieller Code generiert werden, mit dem das reduzierte Modell beispielsweise unter Matlab/Simulink eingebunden werden könnte. Dies würde die Kapselung und einfache Weiterverwendung des reduzierten Modells ergänzen.

Neben der hierarchischen Optimierung könnte eine weitere Verknüpfungsmöglichkeit von Modellierung und Mehrzieloptimierung untersucht werden. Bisher wird die Modellordnungsreduktion nur auf unterlagerte Teilsysteme angewendet. Stattdessen ist es auch denkbar, eine schnell zu berechnende Approximation der Paretomenge mit Hilfe stark reduzierter Modelle zu realisieren und dann zur weiteren Verfeinerung der Optimierungsergebnisse die Modellierungstiefe sukzessive zu erhöhen. Ob und in welcher Form Verfahren der parametrischen Modellordnungsreduktion hierfür eingesetzt werden können, ist zukünftig zu untersuchen.

8 Literaturverzeichnis

- [ABG10] ANTOULAS, A. C.; BEATTIE, C. A.; GUGERCIN, S.: Interpolatory Model Reduction of Large-Scale Dynamical Systems. In: MOHAMMADPOUR, J. (Hrsg.); GRIGORIADIS, K. M. (Hrsg.): *Efficient Modeling and Control of Large-Scale Systems*. Springer, 2010, S. 3–58
- [ADG⁺08] ADEL T, P.; DONOTH, J.; GAUSEMEIER, J.; GEISLER, J.; HENKLER, S.; KAHL, S.; KLÖPPER, B.; KRUPP, A.; MÜNCH, E.; OBERTHÜR, S.; PAIZ, C.; PODLOGAR, H.; PORRMANN, M.; RADKOWSKI, R.; ROMAUS, C.; SCHMIDT, A.; SCHULZ, B.; VÖCKING, H.; WITKOWSKI, U.; WITTING, K.; ZNAMENSHCHYKOV, O.: *Selbstoptimierende Systeme des Maschinenbaus – Definitionen, Anwendungen, Konzepte*. Bd. 234. Paderborn: HNI-Verlagsschriftenreihe, 2008
- [AEH⁺11] ADEL T, P.; ESAU, N.; HÖLSCHER, C.; KLEINJOHANN, B.; KLEINJOHANN, L.; KRÜGER, M.; ZIMMER, D.: Hybrid Planning for Self-Optimization in Railbound Mechatronic Systems. In: NAIK, G. (Hrsg.): *Intelligent Mechatronics*. InTech, 2011, S. 169–194
- [AF10] AMOROCHO, J. P.; FASSBENDER, H.: Time dependent weight functions for the Trajectory Piecewise-Linear approach. In: *16th European Conference on Mathematics for Industry*. Wuppertal, 26.-30. Juli 2010
- [AHHS10] ANTIL, H.; HEINKENSCHLOSS, M.; HOPPE, R.; SORENSEN, D.: Domain decomposition and model reduction for the numerical solution of PDE constrained optimization problems with localized optimization variables. In: *Computing and Visualization in Science* 13 (2010), Nr. 6, S. 249–264
- [AM71] ANDERSON, B. D. O.; MOORE, J. B. ; NEWCOMB, R. W. (Hrsg.): *Linear Optimal Control*. Prentice-Hall, 1971 (Networks Series)
- [Ant05] ANTOULAS, A. C.: *Approximation of Large-Scale Dynamical Systems*. Philadelphia: SIAM, 2005
- [Arn51] ARNOLDI, W. E.: The principle minimized iteration in the solution of the matrix eigenproblem. In: *Quarterly of Applied Mathematics* 9 (1951), S. 17–29

- [ASG01] ANTOULAS, A. C.; SORENSEN, D. C.; GUGERCIN, S.: A survey of model reduction methods for large-scale systems. In: *Contemporary Mathematics, AMS Publications* 280 (2001), S. 193–219
- [Bai02] BAI, Z.: Krylov subspace techniques for reduced-order modelling of large-scale dynamical systems. In: *Applied Numerical Mathematics* 43 (2002), S. 9–44
- [BB09] BAUR, U.; BENNER, P.: Modellreduktion für parametrisierte Systeme durch balanciertes Abschneiden und Interpolation. In: *at - Automatisierungstechnik* 57 (2009), Nr. 8, S. 411–419
- [BB11] BENNER, P.; BREITEN, T.: On \mathcal{H}_2 -model reduction of linear parameter-varying systems. In: *Proceedings on Applied Mathematics and Mechanics (PAMM)* Bd. 11, 2011, S. 805–806
- [BB12] BENNER, P.; BREITEN, T.: Krylov-Subspace Based Model Reduction of Nonlinear Circuit Models Using Bilinear and Quadratic-Linear Approximations. In: GÜNTHER, M. (Hrsg.); BARTEL, A. (Hrsg.); BRUNK, M. (Hrsg.); SCHÖPS, S. (Hrsg.); STRIEBEL, M. (Hrsg.): *Progress in Industrial Mathematics at ECMI 2010*, Springer, 2012, S. 153–159
- [BBBG11] BAUR, U.; BEATTIE, C.; BENNER, P.; GUGERCIN, S.: Interpolatory Projection Methods for Parameterized Model Reduction. In: *SIAM Journal on Scientific Computing* 33 (2011), Nr. 5, S. 2489–2518
- [BD08] BOND, B.N.; DANIEL, L.: Guaranteed stable projection-based model reduction for indefinite and unstable linear systems. In: *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2008, S. 728–735
- [BD10] BREITEN, T.; DAMM, T.: Krylov subspace methods for model order reduction of bilinear control systems. In: *Systems & Control Letters* 59 (2010), Nr. 8, S. 443–450
- [BDD12] BHARDWAJ, P.; DASGUPTA, B.; DEB, K.: Modeling Pareto-Optimal Set Using B-Spline Basis Functions / Kanpur Genetic Algorithms Laboratory. 2012 (KanGAL Report No.2012008). – Forschungsbericht
- [BDG07] BRÜLS, O.; DUYSINX, P.; GOLINVAL, J.-C.: The global modal parameterization for non-linear model-order reduction in flexible multibody dynamics. In: *International Journal for Numerical Methods in Engineering* 69 (2007), S. 948–977
- [BG84] BOLEY, D. L.; GOLUB, G. H.: The Lanczos – Arnoldi algorithm and controllability. In: *Systems & Control Letters* 4 (1984), S. 318–324

- [BG09] BEATTIE, C.; GUGERCIN, S.: Interpolatory projection methods for structure-preserving model reduction. In: *Systems & Control Letters* 58 (2009), Nr. 3, S. 225–232. – ISSN 0167–6911
- [BKLS06] BLANKE, M.; KINNAERT, M.; LUNZE, J.; STAROSWIECKI, M.: *Diagnosis and Fault-Tolerant Control*. 2. Springer Verlag, 2006
- [BKR⁺11] BIELAWNY, D.; KRÜGER, M.; REINOLD, P.; TIMMERMANN, J.; TRÄCHTLER, A.: Iterative Learning of Stochastic Disturbance Profiles Using Bayesian Networks. In: *9th IEEE International Conference on Industrial Informatics (INDIN)*, 2011, S. 443 – 450
- [Boo01] BOOR, C. de ; MARSDEN, J. E. (Hrsg.); SIROVICH, L. (Hrsg.): *Applied Mathematical Sciences*. Bd. 27: *A Practical Guide to Splines*. Springer, 2001
- [BS11] BENNER, P.; SAAK, J.: Efficient balancing-based MOR for large-scale second-order systems. In: *Mathematical and Computer Modeling of Dynamical Systems* 17 (2011), Nr. 2, S. 123–143
- [BSMM08] BECHTHOLD, T.; STRIEBEL, M.; MOHAGHEGH, K.; MATEN, E.J.W. ter: Nonlinear Model Order Reduction in Nanoelectronics: Combination of POD and TPWL. In: *PAMM - Proceedings of Applied Mathematics and Mechanics* Bd. 8, 2008, S. 10057–10060
- [BZS⁺09] BONIN, T.; ZÄH, M.; SOPPA, A.; FASSBENDER, H.; SAAK, J.; BENNER, P.: Modale versus moderne Ordnungsreduktionsverfahren: Effiziente Simulation von Werkzeugmaschinen. In: *MECHATRONIK* 11-12 (2009), S. 46–47
- [Dav66] DAVISON, E. J.: A Method for Simplifying Linear Dynamic Systems. In: *IEEE Transactions on Automatic Control* 11 (1966), January, Nr. 1, S. 93–101
- [Deb99] DEB, K.: *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley, 1999
- [Del08] DELL’AERE, A.: *Numerical Methods for the Solution of Bi-Level Multi-Objective Optimization Problems*, Universität Paderborn, Diss., 2008
- [DFJ00] DELLNITZ, M.; FROYLAND, G.; JUNG, O.: The Algorithms Behind GAIO - Set Oriented Numerical Methods for Dynamical Systems. In: *Ergodic theory, analysis, and efficient simulation of dynamical systems*, Springer, 2000, S. 145–174
- [DH97] DELLNITZ, M.; HOHMANN, A.: A subdivision algorithm for the computation of unstable manifolds and global attractors. In: *Numerische Mathematik* 75 (1997), S. 293–317

- [Doo92] DOOREN, P. van: Numerical Linear Algebra Techniques for Large Scale Matrix Problems in Systems and Control. In: *31st Conference on Decision and Control*, 1992
- [DS11] DRUSKIN, V.; SIMONCINI, V.: Adaptive rational Krylov subspaces for large-scale dynamical systems. In: *Systems & Control Letters* 60 (2011), Nr. 8, S. 546–560
- [DSC⁺04] DANIEL, L.; SIONG, O. C.; CHAY, L. S.; LEE, K. H.; WHITE, J.: A Multiparameter Moment-Matching Model-Reduction Approach for Generating Geometrically Parameterized Interconnected Performance Models. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 23 (2004), Mai, Nr. 5, S. 678–693
- [ECSP⁺11] EID, R.; CASTANE-SELGA, R.; PANZER, H.; WOLF, T.; LOHMANN, B.: Stability-preserving parametric model reduction by matrix interpolation. In: *Mathematical and Computer Modelling of Dynamical Systems* 17 (2011), Nr. 4, S. 319–335
- [Eid09] EID, R.: *Time Domain Model Reduction By Moment Matching*, Technische Universität München, Diss., 2009
- [ESL⁺06] EID, R.; SALIMBAHRAMI, B.; LOHMANN, B.; RUDNYI, E. B.; KORVINK, J. G.: Parametric Order Reduction of Proportionally Damped Second Order Systems / Lehrstuhl für Regelungstechnik, Technische Universität München. 2006. – Forschungsbericht
- [FB07] FENG, L.; BENNER, P.: A Robust Algorithm for Parametric Model order Reduction. In: *Proceedings in Applied Mathematics and Mechanics (PAMM)*, 2007
- [FBG12] FLAGG, G.; BEATTIE, C.; GUGERCIN, S.: Convergence of the Iterative Rational Krylov Algorithm. In: *Systems & Control Letters* 61 (2012), Juni, Nr. 6, S. 688–691
- [Fen05] FENG, L.: Parameter independent model order reduction. In: *Mathematics and Computers in Simulation* 68 (2005), S. 221–234
- [FF95] FELDMANN, P.; FREUND, R. W.: Efficient linear circuit analysis by Padé approximation via the Lanczos process. In: *IEEE Transactions on Computer-Aided Design* 14 (1995), S. 639–649
- [FHIDE06] FARLE, O.; HILL, V.; INGELSTRÖM, P.; DYCZIJ-EDLINGER, R.: Ordnungsreduktion linearer zeitinvarianter Finite-Elemente-Modelle mit multivariabler polynomieller Parametrierung. In: *at - Automatisierungstechnik* 54 (2006), Nr. 4, S. 161–169
- [Föl08] FÖLLINGER, O.: *Regelungstechnik*. Heidelberg: Hüthig Verlag, 2008

- [Fou07] FOURIER, J. B. J.: *On the Propagation of Heat in Solid Bodies*. Submitted to the Paris Institute, 1807
- [Fre03] FREUND, R. W.: Model reduction methods based on Krylov subspaces. In: *Acta Numerica* 12 (2003), S. 267–319
- [GAB08] GUGERCIN, S.; ANTOULAS, A. C.; BEATTIE, C.: \mathcal{H}_2 Model Reduction for Large-Scale Linear Dynamical Systems. In: *SIAM Journal on Matrix Analysis and Applications* 30 (2008), Nr. 2, S. 609–638
- [GFDK08a] GAUSEMEIER, J.; FRANK, U.; DONOTH, J.; KAHL, S.: Spezifikationsstechnik zur Beschreibung der Prinzipiellösung selbstoptimierender Systeme des Maschinenbaus – Teil 1. In: *Konstruktion* 7/8 (2008), Juli/August, S. 59–66. – Springer-VDI-Verlag, Düsseldorf
- [GFDK08b] GAUSEMEIER, J.; FRANK, U.; DONOTH, J.; KAHL, S.: Spezifikationsstechnik zur Beschreibung der Prinzipiellösung selbstoptimierender Systeme des Maschinenbaus – Teil 2. In: *Konstruktion* 9 (2008), September, S. 91–108. – Springer-VDI-Verlag, Düsseldorf
- [Glo84] GLOVER, K.: All-optimal Hankel-norm approximations of linear multivariable systems with \mathcal{L}_∞ error bounds. In: *International Journal of Control* 39 (1984), S. 1115–1193
- [GP11] GRÜNE, L.; PANNEK, J.: *Nonlinear Model Predictive Control*. Springer, 2011
- [GPS11] GOLLER, B.; PRADLWARTER, H. J.; SCHUËLLER, G. I.: An interpolation scheme for the approximation of dynamical systems. In: *Computer Methods in Applied Mechanics and Engineering* 200 (2011), Nr. 1-4, S. 414–423
- [Gra12] GRAICHEN, K.: A fixed-point iteration scheme for real-time model predictive control. In: *Automatica* 48 (2012), S. 1300–1305
- [Gri97] GRIMME, E. J.: *Krylov projection methods for model reduction*, ECE Department, University of Illinois, Urbana-Champaign, Diss., 1997
- [GT09] GEISLER, J.; TRÄCHTLER, A.: Control of the Pareto Optimality of Systems with Unknown Disturbances. In: *International Conference on Control and Automation*. Christchurch, New Zealand, December, 9–11, 2009
- [Gu11] GU, C.: QLMOR: A Projection-Based Nonlinear Model Order Reduction Approach Using Quadratic-Linear Representation of Nonlinear Systems. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 30 (2011), Nr. 9, S. 1307–1320

- [GVD04] GALLIVAN, K.; VANDENDORPE, A.; DOOREN, P. V.: Model Reduction of MIMO Systems via Tangential Interpolation. In: *SIAM Journal on Matrix Analysis and Applications* 26 (2004), Nr. 2, S. 328–349
- [GWTD08] GEISLER, J.; WITTING, K.; TRÄCHTLER, A.; DELLNITZ, M.: Multiobjective Optimization of Control Trajectories for the Guidance of a Rail-bound Vehicle. In: *17th IFAC World Congress*. Seoul, Korea, 6.-11. Juli 2008, S. 4380–4386
- [Hes06] HESTERMEYER, T.: *Strukturierte Entwicklung der Informationsverarbeitung für die aktive Federung eines Schienenfahrzeugs*, Universität Paderborn, Diss., September 2006
- [HHKS08] HENKLER, S.; HIRSCH, M.; KAHL, S.; SCHMIDT, A.: Development of Self-optimizing Systems: Domain-spanning and Domain-specific Models exemplified by an Air Gap Adjustment System for Autonomous Vehicles. In: *Proceedings of 2008 ASME International Design Engineering Technical Conferences (IDETC) and Computers and Information in Engineering Conference (CIE)*. New York, USA, 3.-6. August 2008
- [Hil01] HILLERMEIER, C.: *Nonlinear multiobjective optimization: a generalized homotopy approach*. Basel: Birkhäuser Verlag, 2001
- [HKK⁺12] HÖLSCHER, C.; KESSLER, J. H.; KRÜGER, M.; TRÄCHTLER, A.; ZIMMER, D.: Hierarchical Optimization of Coupled Self-optimizing Systems. In: *10th IEEE International Conference on Industrial Informatics*, 2012
- [HO04] HESTERMEYER, T.; OBERSCHELP, O.: Structured Information Processing for Self-Optimizing Mechatronic Systems. In: *1st International Conference on Informatics in Control, Automation and Robotics*. Setubal, Portugal, 2004
- [HPB12] HEINZEMANN, C.; PRIESTERJAHN, C.; BECKER, S.: Towards Modeling Reconfiguration in Hierarchical Component Architectures. In: *Proceedings of the 15th ACM SigSoft International Symposium on Component-Based Software Engineering (CBSE 2012)*, ACM, 2012, S. 23–28
- [HPZA10] HEHENBERGER, P.; POLTSCHAK, F.; ZEMAN, K.; AMRHEIN, W.: Hierarchical design models in the mechatronic product development process of synchronous machines. In: *Mechatronics* 20 (2010), S. 864–875

- [HRA11] HEINKENSCHLOSS, M.; REIS, T.; ANTOULAS, A. C.: Balanced truncation model reduction for systems with inhomogeneous initial conditions. In: *Automatica* 47 (2011), S. 559–564
- [HTS⁺08] HENKE, C.; TICHY, M.; SCHNEIDER, T.; BÖCKER, J.; SCHÄFER, W.: System Architecture and Risk Management for Autonomous Railway Convoys. In: *IEEE International Systems Conference*. Montreal, Canada, 7.-10. April 2008
- [Ise08] ISERMANN, R.: *Mechatronische Systeme*. Springer, 2008
- [Jan10] JANSCHKE, K.: *Systementwurf mechatronischer Systeme*. Springer, 2010
- [Kar39] KARUSH, W.: *Minima of Functions of Several Variables with Inequalities as Side Conditions*. Chicago, USA, Department of Mathematics, University of Chicago, Diplomarbeit, 1939
- [Kas85] KASPER, R.: *Entwicklung und Erprobung eines instrumentellen Verfahrens zum Entwurf von Mehrgrößenregelungen*, Universität Paderborn, Diss., 1985
- [KDL08] KOCH, G.; DIEPOLD, K. J.; LOHMANN, B.: Multi-Objective Road Adaptive Control of an Active Suspension System. In: *Motion and Vibration Control – Selected papers from MOVIC 2008*. Berlin: Springer Verlag, 2008
- [Kie86] KIENDL, H.: Das Konzept der invarianten Ordnungsreduktion. In: *Automatisierungstechnik* 34 (1986), Nr. 12, S. 465–473
- [KP88] KIENDL, H.; POST, K.: Invariante Ordnungsreduktion mittels transparenter Parametrierung. In: *Automatisierungstechnik* 36 (1988), Nr. 3, S. 92–101
- [KPL09] KOCH, G.; PELLEGRINI, E.; LOHMANN, B.: Zustandsadaptive Regelung eines aktiven Fahrwerks - Entwurf und Implementierung am Viertelfahrzeugprüfstand. In: GAUSEMEIER, J. (Hrsg.); RAMMIG, F. (Hrsg.); SCHÄFER, W. (Hrsg.); TRÄCHTLER, A. (Hrsg.): *6. Paderborner Workshop Entwurf mechatronischer Systeme*. Paderborn: HNI Verlagsschriftenreihe, April 2009, S. 43–56
- [Krü09] KRÜGER, M.: Parametrische Modellreduktion für selbstoptimierende mechatronische Systeme. In: LOHMANN, B. (Hrsg.); KUGI, A. (Hrsg.) ; VDI/VDE-GMA (Veranst.): *Tagungsband Workshop GMA-Fachausschuß 1.30/1.40*. Salzburg: Technische Universität Wien, 2009

- [KRKT13] KRÜGER, M.; REMIREZ, A.; KESSLER, J. H.; TRÄCHTLER, A.: Objective-based Control for Self-Optimizing Systems. In: *American Control Conference (ACC)*. Washington, DC, USA, 17.-19. Juni 2013
- [KST10] KRÜGER, M.; SCHARFENBAUM, I.; TRÄCHTLER, A.: Parametrische Modellreduktion in hierarchisch modellierten selbstoptimierenden Systemen. In: GAUSEMEIER, J. (Hrsg.); RAMMIG, W. F. and S. F. and Schäfer (Hrsg.); TRÄCHTLER, A. (Hrsg.) ; Heinz Nixdorf Institut (Veranst.): *7. Paderborner Workshop Entwurf mechatronischer Systeme* Heinz Nixdorf Institut, HNI Verlagsschriftenreihe, Paderborn, 18.-19. März 2010, S. 203–218
- [KSWRV09] KLÖPPER, B.; SONDERMANN-WÖLKE, C.; ROMAUS, C.; VÖCKING, H.: Probabilistic Planning Integrated in a Multi-level Dependability Concept for Mechatronic Systems. In: *IEEE Symposium on Computational Intelligence in Control and Automation*, 2009, S. 104–111
- [KT51] KUHN, H. W.; TUCKER, A. W.: Nonlinear Programming. In: NEYMAN, J. (Hrsg.): *Proceedings of 2nd Berkeley Symposium on Mathematical Statistics and Probability*. Berkeley: University of California Press, 1951, S. 481–492
- [KT12] KRÜGER, M.; TRÄCHTLER, A.: Approximation of Pareto-Optimal Systems using Parametric Model-Order Reduction. In: *7th Vienna International Conference on Mathematical Modelling (MATHMOD)*, 2012
- [Kub08] KUBALINSKA, D.: *Optimal interpolation-based model reduction*, Universität Bremen, Diss., September 2008
- [KWTD11] KRÜGER, M.; WITTING, K.; TRÄCHTLER, A.; DELLNITZ, M.: Parametric Model-Order Reduction in Hierarchical Multiobjective Optimization of Mechatronic Systems. In: *18th IFAC World Congress*. Milano, Italy, 28. August–2. September 2011
- [Lan50] LANCZOS, C.: An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. In: *Journal of Research of the National Bureau of Standards* 45 (1950), S. 225–280
- [Lan52] LANCZOS, C.: Solution of systems of linear equation by minimized iteration. In: *Journal of Research of the National Bureau of Standards* 9 (1952), S. 33–53
- [LBS09] LI, Y.-T.; BAI, Z.; SU, Y.: A two-directional Arnoldi process and its application to parametric model order reduction. In: *Journal of Computational and Applied Mathematics* 226 (2009), S. 10–21

- [LHLH01] LÜCKEL, J.; HESTERMEYER, T.; LIU-HENKE, X.: Generalization of the Cascade Principle in View of a Structured Form of Mechatronic Systems. In: *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. Villa Olmo, Como, Italy, 8.-11. Juli 2001
- [Lit79] LITZ, L.: Ordnungsreduktion linearer Zustandsraummodelle durch Beibehaltung der dominanten Eigenbewegungen. In: *Regelungstechnik* 27 (1979), S. 80–86
- [LK05] LEUNG, A. T.-M.; KHAZAKA, R.: Parametric Model Order Reduction Technique For Design Optimization. In: *Proceedings of International Symposium on Circuits and Systems*, 2005, S. 1290–1293
- [LLL⁺05] LI, P.; LIU, F.; LI, X.; PILEGGI, L. T.; NASSIF, S. R.: Modeling Interconnect Variability Using Efficient Parametric Model Order Reduction. In: *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition*. München, 2005, S. 958–963
- [LS04] LOHMANN, B.; SALIMBAHRAMI, B.: Ordnungsreduktion mittels Krylov-Unterraummethoden. In: *at - Automatisierungstechnik* 52 (2004), Nr. 1, S. 30–38
- [MAK⁺08] MÜNCH, E.; ADELTE, P.; KRÜGER, M.; KLEINJOHANN, B.; TRÄCHTLER, A.: Hybrid Planning and Hierarchical Optimization of Mechatronic Systems. In: *International Conference on Control, Automation and Systems*. Seoul, Korea, 14.-17. Oktober 2008
- [Mat11] MATHWORKS: *MATLAB - Curve Fitting Toolbox*. <http://www.mathworks.de/products/curvefitting/>. Version: 3.2 (2011)
- [Mie99] MIETTINEN, K.: *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, 1999
- [Mün12] MÜNCH, E.: *Selbstopтимierung verteilter mechatronischer Systeme auf Basis paretooptimaler Systemkonfigurationen*, Universität Paderborn, Diss., Dezember 2012
- [Moo81] MOORE, B. C.: Principal Component Analysis in Linear Systems – Controllability, Observability, and Model Reduction. In: *IEEE Transactions on Automatic Control* 26 (1981), Nr. 1, S. 17–32. – IEEE Control System Society, Boston
- [NBP13] NBP – NEUE BAHNTECHNIK PADERBORN: *RailCab*. <http://www.railcab.de/>, 2013
- [OC98] ODABASIOGLU, A.; CELIK, M.: PRIMA: Passive Reduced-Order Interconnect Macromodeling Algorithm. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 17 (1998), Nr. 8, S. 645–654

- [OT08] OSMIC, S.; TRÄCHTLER, A.: Flatness-based Online Controller Reconfiguration. In: *34th Annual Conference of the IEEE Industrial Electronics Society (IECON'08)*. Orlando, Florida, USA, 2008
- [Pad92] PADÉ, H.: Sur la représentation approchée d'une fonction par des fractions rationnelles. In: *Annales scientifiques de l'École Normale Supérieure* 9 (1892), S. 3–93
- [PBFG05] PAHL, G.; BEITZ, W.; FELDHOUSEN, J.; GROTE, K.-H.: *Konstruktionslehre. Grundlagen erfolgreicher Produktentwicklung. Methoden und Anwendung*. 6. Springer-Verlag Berlin Heidelberg, 2005
- [PMEL10] PANZER, Heiko; MOHRING, Jan; EID, Rudy; LOHMANN, Boris: Parametric Model Order Reduction by Matrix Interpolation. In: *at – Automatisierungstechnik* 58 (2010), S. 475–484
- [PR90] PILLAGE, L. T.; ROHRER, R. A.: Asymptotic waveform evaluation for timing analysis. In: *IEEE Trans. Computer-Aided Design* 9 (1990), S. 352–366
- [Pri09] PRIESTERJAHN, C.: Hazard Analysis of Self-Optimizing Mechatronic Systems:. In: *Proc. of the Doctoral Symposium of the 7th joint meeting of the European Software Engineering Conference (ESEC) and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE)*. Amsterdam, 24.-28. August 2009
- [RMG⁺06] RUDNYI, E. B.; MOOSMANN, C.; GREINER, A.; BECHTOLD, T.; KORVINK, J. G.: Parameter Preserving Model Reduction for MEMS System-level Simulation and Design. In: *5th Vienna International Conference on Mathematical Modelling (MATHMOD)*. Wien, Österreich, 8.-10. Februar 2006
- [Rop09] ROPPENECKER, G.: Zustandsregelung linearer Systeme - Eine Neubetrachtung. In: *at – Automatisierungstechnik* 57 (2009), Nr. 10, S. 491–498
- [RW03] REWIENSKI, M.; WHITE, J.: A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices. In: *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* 22 (2003), Februar, Nr. 2, S. 155–170
- [Sal05] SALIMBAHRAMI, B.: *Structure Preserving Order Reduction of Large Second Order Systems using Krylov Subspace Methods*, Technische Universität München, Diss., 2005
- [SBS⁺09] SILVA, Maíra M.; BRÜLS, Olivier; SWEVERS, Jan; DESMET, Wim; BRUSSEL, Hendrik V.: Computer-aided integrated design for ma-

- chines with varying dynamics. In: *Mechanism and Machine Theory* 44 (2009), S. 1733–1745
- [Sch04] SCHÜTZE, O.: *Set Oriented Methods for Global Optimization*, University of Paderborn, Germany, Diss., 2004
- [Sch06] SCHLAUTMANN, Philipp: *Entwicklung eines neuartigen dreidimensionalen aktiven Federungssystems für ein Schienenfahrzeug*, Lehrstuhl für Regelungstechnik und Mechatronik, Universität Paderborn, Diss., 2006
- [Sch09] SCHARFENBAUM, I.: *Parametrische Modellreduktion mechatronischer Systeme durch impliziten Momentenabgleich*, Universität Paderborn, Diplomarbeit, September 2009
- [sfb13] *Sonderforschungsbereich 614 – Selbstoptimierende Systeme des Maschinenbaus der Universität Paderborn*. <http://www.sfb614.de>, 2013
- [SK11] SCHWARZ, H. R.; KÖCKLER, N.: *Numerische Mathematik*. 8. Springer Vieweg, 2011
- [SKW96] SILVEIRA, M. L.; KAMON, M.; WHITE, J.: Efficient reduced-order modeling of frequency-dependent coupling inductances associated with 3-D interconnect structures. In: *Components, Packaging, and Manufacturing Technology, Part B: Advanced Packaging, IEEE Transactions on* 19 (1996), Mai, Nr. 2, S. 283–288
- [SLBK05] SALIMBAHRAMI, B.; LOHMANN, B.; BECHTHOLD, T.; KORVINK, J. G.: A two-sided Arnoldi algorithm with stopping criterion and MIMO selection procedure. In: *Mathematical and Computer Modelling of Dynamical Systems* 11 (2005), Nr. 1, S. 79–93
- [Ste05] STEFFEN, T.: *Control Reconfiguration of Dynamical Systems*. Springer Verlag, 2005
- [SVR08] SCHILDERS, W. H. A. (Hrsg.); VORST, H. A. d. (Hrsg.); ROMMES, J. (Hrsg.): *Model Order Reduction – Theory, Research Aspects and Applications*. Berlin: Springer, 2008 (Mathematics in Industry 13)
- [SW10] SCHEIDL, R.; WINKLER, B.: Model relations between conceptual and detail design. In: *Mechatronics* 20 (2010), S. 842–849
- [SZ05] SCHMIDT, A.; ZIMMER, D.: Der Luftspalt bei Linearmotorgetriebenen Schienenfahrzeugen. In: *Antriebstechnik* 2 (2005)
- [TG11] TAYEB, S.; GIVOLI, D.: Optimal modal reduction of dynamic subsystems: Extensions and improvements. In: *International Journal for Numerical Methods in Engineering* 85 (2011), S. 1–30

- [Thy13] THYSSENKRUPP TRANSRAPID GMBH: *Transrapid*. <http://www.transrapid.de/>, 2013
- [TMV06] TRÄCHTLER, A.; MÜNCH, E.; VÖCKING, H.: Iterative Learning and Self-Optimization Techniques for the Innovative Railcab-System. In: *32nd Annual Conference of the IEEE Industrial Electronics Society (IECON)*. Paris, France, 2006, S. 4683–4688
- [Trä06] TRÄCHTLER, A.: Railcab - mit innovativer Mechatronik zum Schienenverkehrssystem der Zukunft. In: *VDE-Kongress*. Aachen, Deutschland, 23.-25. Oktober 2006
- [VDI04a] *VDI 2057, Part 1: "Human exposure to mechanical vibrations – Whole-body vibration"*. Beuth Verlag, Berlin, 2004
- [VDI04b] *VDI 2206 Entwicklungsmethodik für mechatronische Systeme*. VDI-Gesellschaft Produkt- und Prozessgestaltung, Beuth Verlag GmbH, April 2004
- [VRW03] VASILYEV, D.; REWIENSKI, M.; WHITE, J.: A TBR-based Trajectory Piecewise-Linear Algorithm for Generating Accurate Low-order Models for Nonlinear Analog Circuits and MEMS. In: *Design Automation Conference 0* (2003), S. 490–495
- [VT08] VÖCKING, H.; TRÄCHTLER, A.: Self-optimization of an Active Suspension System Regarding Energy Requirements. In: *International Conference on Control, Automation and Systems*. Seoul, Korea, 14.-17. Oktober 2008
- [Wer92] WERNER, J.: *Numerische Mathematik 1*. Vieweg Studium, 1992
- [Wer05] WERNER, Dirk: *Funktionalanalysis*. Springer, 2005
- [WM01] WEILE, D. S.; MICHIELSEN, E.: Analysis of Frequency Selective Surfaces Using Two-Parameter Generalized Rational Krylov Model-Order Reduction. In: *IEEE Transactions on Antennas and Propagation* 49 (2001), Nr. 11, S. 1539–1549
- [WMGG99] WEILE, D. S.; MICHIELSEN, E.; GRIMME, E.; GALLIVAN, K.: A Method for Generating Rational Interpolant Reduced Order Models of Two-Parameter Linear Systems. In: *Applied Mathematics Letters* 12 (1999), S. 93–102
- [WSW06] WITTIG, T.; SCHUHMANN, R.; WEILAND, T.: Model order reduction for large systems in computational electromagnetics. In: *Linear Algebra and its Applications* 415 (2006), Nr. 2-3, S. 499–530. – Elsevier

-
- [Wul04] WULFF, K.: *Quadratic and Non-Quadratic Stability Criteria for Switched Linear Systems*. Maynooth, National University of Ireland, Diss., Dezember 2004
- [XZ11] XU, Y.; ZENG, T.: Optimal \mathcal{H}_2 Model Reduction for Large-Scale MIMO Systems via Tangential Interpolation. In: *International Journal of Numerical Analysis and Modeling* 8 (2011), Nr. 1, S. 174–188
- [YM12] YUE, Y.; MEERBERGEN, K.: Using Krylov-Padé model order reduction for accelerating design optimization of structures and vibrations in the frequency domain. In: *International Journal for Numerical Methods in Engineering* 90 (2012), Nr. 10, S. 1207–1232